START HERE

PixelClaw customer guide

Choose a plan, publish your first article and connect your blog website to the places readers already visit.

01 Check the current plan

Open **Pricing** and compare **Monthly** and **Yearly**. Prices, included features and numeric allowances can change. Use the current pricing page and your account's **Billing** screen for the latest offer.

02 Understand Blog Sites

A Blog Site is one separate blog website. Your account's Blog Sites share its credits and numeric plan limits. **Posts kept** counts saved originals, including drafts. Linked language editions share their original's slot. This count does not reset when your credits renew.

03 Check how a post counted

As an owner or admin, open **Billing** and **Usage**. A new original created using any purchased pack credits counts as a paid extra. Buying a pack or editing an existing post leaves its original classification unchanged. Ask Support about an older post whose creation charge is unclear.

Page-view fair use

Page views are shared across the account and reset at the start of each UTC calendar month. Going past the allowance does not block the blog or automatically charge extra. Media storage measures files currently stored and does not reset monthly.

Find your task

02 Publish a post
03 Use your existing website
04 Connect a custom domain
05 Change the blog layout
06 Use the AI API
07 Connect Search Console

Current pricing Product features

WRITING AND PUBLISHING

Publish your first post

Start with one specific question your readers ask. You can write the answer yourself or review an AI-assisted draft before publishing.

01 Create your blog

Sign in. On the blog creation screen, enter a **Blog name** and choose an available **Blog address**. Select **Create blog**. Your hosted address follows the pattern **yourname.pixelclaw.ai**.

02 Choose how to write

Open **Write a post** and enter a topic, such as “How do I choose running shoes for my first 5K?” Choose **Write it myself** for a blank draft. For AI, select a research option, check the displayed credit price and choose **Research and write with AI**.

03 Review the draft

On the AI path, edit the outline headings, order and directions before requesting the draft. Read the article, verify claims against its sources and select **Save draft**. Open **Optimize** to address findings. Use **Preview** to read the saved version.

04 Publish and open the live article

Open **Publish** and select **Publish now**. If the publish check holds the post, fix the listed issues and save again. Open the public article from the confirmation or dashboard and check its title, image and links.

Updating a live article

Save the revised draft, preview it, then select **Publish changes**. To keep a new post private, stop after saving the draft.

WEBSITE INTEGRATION

Use your existing website

Choose where readers should see your posts. A menu link opens the PixelClaw blog. A server-side integration displays articles within pages built by your own website.

01 Add a Blog link without code

In PixelClaw, select **View your site**. Copy the blog's public address and check that a published post is visible. In your website builder or CMS, open the navigation menu editor, add a link labeled **Blog**, paste the public address and publish the menu.

02 For posts on your own pages, create a read key

If your plan includes API access, a blog owner or admin can open **API keys**. Enter a **Key name** for your website, select **Read posts** (**posts:read**) and choose **Create key**. Store the key as a server-side secret. For a public blog, leave preview and write permissions off.

03 Fetch the articles from your server

Have your developer send **GET /api/v1/posts** to PixelClaw with the key in the Authorization header as a Bearer token. Render the published results on your site, and use the **API reference** link in **API keys** for post-detail routes and response fields.

04 Test as a visitor

Open your website in a private browser window. Follow **Blog**, open an article and check its links. For the API option, confirm your site links to the intended article and that the key stays out of page HTML and browser JavaScript.

Want a branded address?

To keep PixelClaw hosting while sending readers to an address such as **blog.example.com**, connect that address first. Follow the steps on page 4.

HOSTING

Connect a custom domain

Use a subdomain such as blog.example.com if your main domain already serves a website. Have access to your DNS provider and an owner or admin role in PixelClaw.

01 Add and confirm the hostname

Open **Settings** → **Publishing** → **Custom domain and DNS**. Enter your hostname under **Blog address** and select **Add address**. If asked to confirm ownership, add the TXT record shown at your DNS provider, then select **Verify DNS and add address**.

02 Create the DNS record

At your DNS provider, add the CNAME shown on the address card. Copy **Name / Host** and **Value / Target** exactly. Leave TTL at Auto or its default. If you use Cloudflare, set this record to **DNS only**.

03 Check DNS and finish HTTPS

Back on the address card, select **Check connection**. If **Finish HTTPS setup** appears, add the additional records PixelClaw shows and check again. Open the HTTPS address once the status is **Live**.

04 Use the address in public links

Select **Make primary** if this should be the address in blog links and the sitemap. Open a published article from the blog homepage and confirm the hostname in your browser.

If the card is still pending

Waiting for DNS means PixelClaw has not found the expected record. Compare both fields and expand **DNS setup help** for hostname formatting. **Setting up HTTPS** means certificate setup is underway. For **Needs attention**, follow the next step on the card.

APPEARANCE

Change the blog layout

The homepage controls how readers browse posts. The article layout controls how they read an individual post. Owners and admins can change blog-wide settings.

01 Choose a homepage

Open **Settings** and select **Appearance**. In **Blog homepage and menu**, select **Magazine** for a grid of story covers or **Journal** for a compact list of headlines.

02 Build the header menu

Choose **Add menu link**, enter a **Menu label** and a **Link address**, and arrange your links. You can add up to six. Select **Save homepage and menu**, then **View public blog** to check your changes. The blog name always links home.

03 Choose the default article layout

If your plan includes post layouts, find **Post layout** and **Default layout**. Select **Guide** for tutorials and detailed articles, or **Editorial** for stories and updates. Your selection saves when you choose it. New posts and Autopilot use this default.

04 Give one article its own layout

Open that post's **Preview** step. Use the layout control to choose the design for this article and check the preview. Publish the post or choose **Publish changes** for an existing article.

When readers see the change

Saving the homepage and menu updates the public homepage immediately. An already published post keeps its article layout until you publish it again.

DEVELOPER WORKFLOW

Use the AI API

If your plan includes AI over the API, your own software can ask PixelClaw to research, outline or draft an article.

01 Create a scoped PixelClaw key

As an owner or admin, open **API keys**. Name the key and choose **Read AI models, credits and jobs** (**ai:read**) and **Start AI work** (**ai:write**). Select **Create key**. Store the secret on your server and send it as a Bearer token.

02 Check the model and balance

Requests use the PixelClaw API base path **/api/v1**. Read **GET /ai/models** for available models and costs, and **GET /credits** for the remaining balance. Open **API reference** from **API keys** for the request body required by each job.

03 Start a job and follow its result

Send **POST /ai/jobs** with the job input and an **Idempotency-Key** header. Choose a new value for each intended job. Reuse that value if you retry the same request. Read **GET /ai/jobs/{id}** using the returned job ID until it completes or fails.

04 Carry each result into the next step

For a full post, start a **research** job. Pass its **research_id** to an **outline** job. Pass the returned **outline_id** to a **draft** job. Use its **post_id** to locate the draft and review it in PixelClaw before publishing.

Using your own provider account

Under **Settings** → **Integrations** → **Writing model and billing**, you can use a saved writing key when the available balance cannot cover a whole step. That step runs on the eligible key. Web lookups and images have separate billing choices.

ANALYTICS

Connect Search Console

See how often Google shows your blog and which searches lead to clicks. Use a plan with Analytics and ask a blog owner or admin to connect the account.

01 Verify your blog with Google

For a URL-prefix property, enter the exact public HTTPS address. For Google's HTML tag method, open **Settings** → **Publishing** → **Crawlers and search engines** → **Prove the site is yours**. Paste the tag into **Google code**, choose **Save codes**, then verify in Google. Domain properties use DNS verification.

02 Connect the Google account

Open **Analytics** and find **Google Search Console**. Select **Connect Search Console** and sign in to the same Google account that can access the verified property.

03 Select the matching property

Choose the property that covers the blog address and select **Use this property**. If none appears, verify the address in the correct Google account, return to PixelClaw and select **Check again**. Use **Change property** to switch later.

04 Read the first report

Return to Analytics to review impressions, clicks and searches for the last 28 completed days. Google's figures usually arrive about two days late. A newly published blog may take several weeks to appear in search.

Optional sitemap submission

If your plan offers it, use **Allow sitemap sending** and approve the requested Google permission. Select **Send sitemap to Google** and check the last submission status. Google decides which submitted pages to index.