

# Blog

[Standalone blog engine](#) · [Posts](#) · [Categories](#) · [Tags](#) · [Authors](#) · [Comments](#) · [Luma + Hyvä](#)

A fast, SEO-friendly, fully self-contained blog for Magento 2 / Adobe Commerce. It owns all of its own data and rendering — posts, categories, tags, authors, threaded comments, a table of contents, reading time, RSS, JSON-LD structured data and product-page integration — with no third-party blog engine required, on both Luma and Hyvä.

## **Etechflow\_Blog** · **Module Documentation & User Guide**

Compatible with Magento / Adobe Commerce 2.4.6 – 2.4.9 · PHP 8.1 – 8.4 · Part of the Etechflow Suite

© Etechflow · [license-service.etechflow.com](https://license-service.etechflow.com)

# Table of Contents

1. Overview
2. Architecture
3. Installation & Setup
4. Configuration — General
5. Managing Content (Admin)
6. Comment Moderation
7. Storefront Experience
8. SEO, Feeds & Discovery
9. Product-Page Integration
10. Configuration Reference
11. Licensing & Activation
12. Troubleshooting / FAQ

## 1 • Overview

### 1.1 What the module does

Etechflow Blog adds a complete, self-contained **blog** to your Magento 2 / Adobe Commerce storefront. It is a full content platform — **posts, nested categories, tags, authors and threaded comments** — rendered as fast, server-side pages that work identically on **Luma and Hyvä**. Every post gets an automatic **table of contents, reading time, a views counter, social share** buttons and SEO metadata (canonical, Open Graph, Twitter cards and JSON-LD). Content is managed from a dedicated admin area; nothing depends on a third-party blog extension.

### 1.2 Highlights

- **Standalone** — owns its own data and logic; no third-party blog engine required.
- **Theme-agnostic** — self-contained scoped CSS (no Tailwind or theme LESS dependency), so it looks the same on Luma and Hyvä.
- **Full entity set** — Posts, nested Categories, Tags, Authors and moderated Comments, each with its own admin grid and form.
- **Reader features** — auto table of contents, reading time, reading-progress bar, views counter, social share, related posts and next/prev.
- **SEO-first** — per-entity meta, self-referencing canonical, Open Graph / Twitter cards, **JSON-LD** (BlogPosting, BreadcrumbList, Person, Organization), XML sitemap and an **RSS** feed.
- **Discovery** — sidebar widgets (search, categories, recent, popular, archive, RSS), a storefront **top-menu link** and a licence-aware **footer link**.
- **Catalog integration** — a “Related articles” block on product pages links posts to the products they discuss.
- **Licensed** — activation via the eTechFlow portal; the blog and its admin lock down when unlicensed.

### 1.3 Compatibility matrix

Area	Supported
Magento / Adobe Commerce	2.4.6 - 2.4.9 (tested on 2.4.8)
PHP	8.1 - 8.4
Themes	Luma, Hyvä (and Adobe Commerce themes)
Front-end tech	Server-rendered PHTML + scoped CSS + tiny vanilla JS (no jQuery / RequireJS / Alpine dependency)
Licensing	eTechFlow license-service (SP / HMAC / Bundle keys)

## 2 • Architecture

### 2.1 The module

A single self-contained module — **Etechflow\_Blog** (package `etechflow/module-etechflow-blog`). It ships a storefront route, a clean-URL router, admin grids and forms for every entity, a configuration section, sidebar widgets, a product-page block, an RSS feed, and a portal-based licence layer. All blog data lives in its own `etechflow_blog_*` tables — it never touches the catalog or any other blog extension.

### 2.2 How it works

- **Storefront route** — a configurable base path (default `/blog`) renders the blog home, post, category, tag, author, archive and search pages via a clean-URL router.
- **Entities** — Posts, Categories (nested), Tags, Authors and Comments, each stored in its own table and managed from its own admin grid + form.

- **Rendering** — content is server-rendered PHTML with self-contained `etf--`-prefixed CSS, so it is fully visible without JavaScript and looks identical on Luma and Hyvä.
- **SEO layer** — a head block emits canonical, Open Graph, Twitter and JSON-LD tags; a sitemap provider and an RSS feed expose the content to search engines and readers.
- **Licence chokepoint** — a single `LicenseValidator::isValid()` decision drives a storefront `FrontendGate` (whole blog route → 404 when unlicensed) and an admin `AdminGate` (blog admin → licence gate).

### 2.3 The single licence chokepoint

Everything pivots on `LicenseValidator::isValid()`. When the licence is missing, suspended, expired or bound to a different domain, a **FrontendGate** observer returns the entire blog route as a **404**, and an **AdminGate** observer redirects the blog admin pages to the in-admin licence gate. The storefront top-menu and footer links, and the product-page “Related articles” block, all hide themselves in the same condition — so an unlicensed blog leaves no dangling links.

## 3 • Installation & Setup

### 3.1 Install & enable

Distributed via the private eTechFlow Composer repository; your licence key doubles as the download token. From your Magento project root:

```
composer config repositories.etechflow composer
https://license-service.etechflow.com/composer

composer config http-basic.license-service.etechflow.com you@example.com SP-XXXX-XXXX-XXXX-XXXX

composer require etechflow/module-etechflow-blog

bin/magento setup:db-declaration:generate-whitelist --module-name=Etechflow_Blog

bin/magento module:enable Etechflow_Blog

bin/magento setup:upgrade

bin/magento setup:di:compile          # production mode

bin/magento setup:static-content:deploy en_GB # your store locale(s)

bin/magento cache:flush
```

#### Important

In **production mode**, always run `setup:static-content:deploy` for your store’s actual locale (e.g. `en_GB`), not just `en_US` — otherwise the storefront cannot find the blog’s CSS and the pages render unstyled. And after any configuration change, **flush the cache**: saving config only invalidates the Full Page Cache in production, it does not rebuild it.

### 3.2 Where it lives

Action	Path
--------	------

<b>Blog admin</b>	Admin → Etechflow Blog (Posts, Categories, Tags, Authors, Comments)
<b>Configuration</b>	Stores → Configuration → Etechflow → Blog
<b>Licence</b>	Etechflow Blog → License
<b>Storefront</b>	/blog (or your configured base path, e.g. /etechflow)
<b>RSS feed</b>	/blog/rss



Figure 1 — The Etechflow Blog admin menu — License, Posts, Categories, Tags, Authors, Comments and Configuration. Everything you manage lives under this one menu.

## 4 • Configuration — General

Open **Stores** → **Configuration** → **Etechflow** → **Blog**. The **General** group holds the settings that shape every blog page: whether the blog is on, the base URL path, the blog title, how many posts appear per listing page, the excerpt behaviour, and the storefront navigation link.

General

**Master switch** → **Enable Blog** (store view) Yes

Master on/off switch. When set to No the whole blog (all /blog URLs) becomes unavailable.

**URL segment** → **Blog Base URL Path** (store view) blog

The first part of every blog URL, e.g. "blog" gives /blog, /blog/my-post. Use letters/numbers/hyphens only. Flush cache after changing.

**Blog Title** (store view) Blog

Shown as the H1 / page title on the blog home page and used in the breadcrumb.

**Pagination** → **Posts Per Page** (store view) 9

How many posts appear on each listing page before pagination kicks in. Recommended 9-12.

**Show Excerpt In Listings** (store view) Yes

Show a short preview of each post in listing grids. If a post has no manual excerpt, one is generated automatically.

**Auto Excerpt Length (characters)** (store view) 200

Maximum length of an auto-generated excerpt when a post has no manual short content.

**Add Blog Link To Top Menu** (store view) Yes

Adds a link to the blog in the storefront main navigation menu.

**Top Menu Link Text** (store view) Blog

The wording of the menu link, e.g. "Blog", "News", "Guides".

Design

Figure 2 — General configuration (defaults). Enable Blog, Blog Base URL Path (blog), Blog Title, Posts Per Page (9), Show Excerpt In Listings, Auto Excerpt Length (200 chars) and the Top-Menu link options — each with an inline hint.

Each setting is documented inline. **Enable Blog** is the master switch. **Blog Base URL Path** sets the first URL segment — `blog` gives `/blog` and `/blog/my-post`; you can rename it to anything (letters, numbers, hyphens). **Blog Title** is the H1 and breadcrumb label of the blog home. **Posts Per Page** controls pagination; **Show Excerpt In Listings** and **Auto Excerpt Length** control the preview text; and **Add Blog Link To Top Menu / Top Menu Link Text** place a link to the blog in the main storefront navigation.

### 4.1 Changing the base path and title — worked example

To show that these settings drive the live pages, here the **Blog Base URL Path** is changed to `etechflow` and the **Blog Title** to `EtechFlowBlog`:

## General

Enable Blog (store view) Yes

Master on/off switch. When set to No the whole blog (all /blog URLs) becomes unavailable.

Blog Base URL Path (store view) etechflow ← Custom URL path

The first part of every blog URL, e.g. "blog" gives /blog, /blog/my-post. Use letters/numbers/hyphens only. Flush cache after changing.

Blog Title (store view) EtechFlowBlog ← Custom blog title

Shown as the H1 / page title on the blog home page and used in the breadcrumb.

Posts Per Page (store view) 9

How many posts appear on each listing page before pagination kicks in. Recommended 9-12.

Show Excerpt in Listings (store view) Yes

Show a short preview of each post in listing grids. If a post has no manual excerpt, one is generated automatically.

Figure 3 — General configuration customised — Blog Base URL Path set to “etechflow” and Blog Title set to “EtechFlowBlog”.

After saving **and flushing the cache**, the storefront reflects both changes at once — the blog now lives at /etechflow and the page title / breadcrumb read **EtechFlowBlog**:

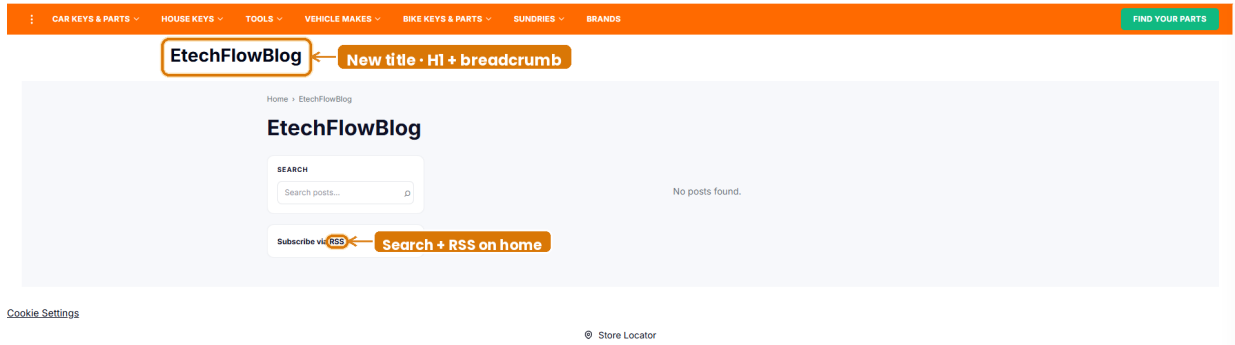


Figure 4 — The storefront blog home after the change — the new title “EtechFlowBlog” in the H1 and breadcrumb, served from the renamed /etechflow path.

### Note

The base-path field’s own hint says “Flush cache after changing.” On a production store a config save invalidates the page cache but does not rebuild it, so the change only appears after a cache flush (and a browser hard-refresh). This is deploy-mode behaviour, not the module.

## 5 • Managing Content (Admin)

All content is created and edited from the **Etechflow Blog** admin menu. Categories organise posts; posts are the articles themselves; tags and authors enrich them. Each entity has a standard Magento grid (search, filter, columns, mass actions) and a full edit form.

### 5.1 Categories

Categories group related posts and produce category pages (/blog/<category>). Manage them at **Etechflow Blog → Categories**.

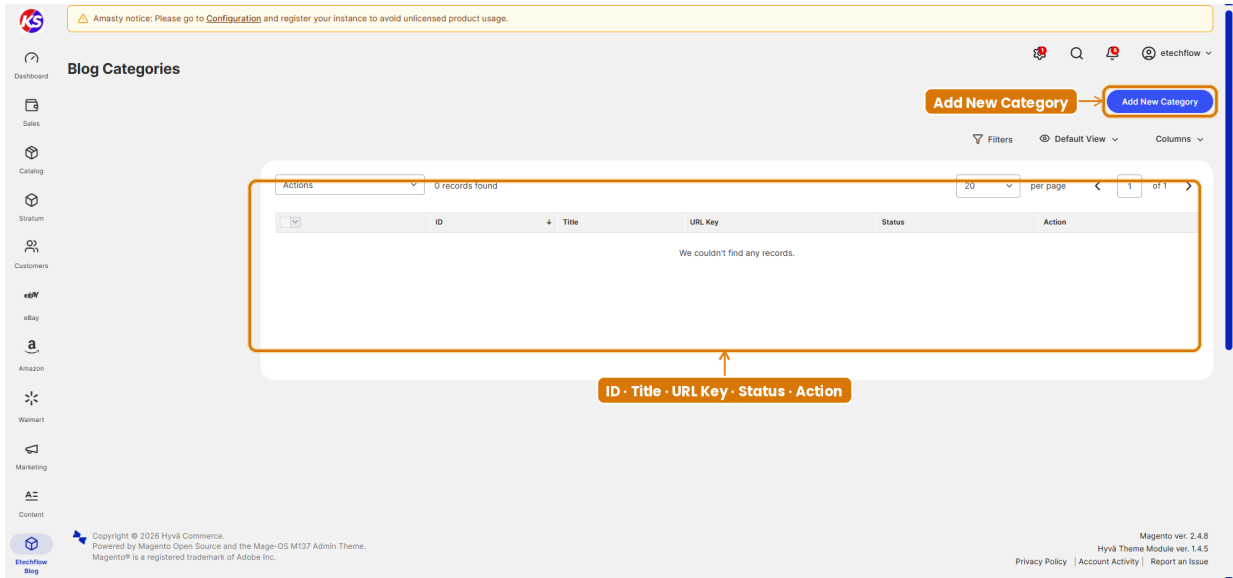


Figure 5 — The Blog Categories grid — ID, Title, URL Key, Status and Action columns, with Add New Category, Filters and column controls. (Shown empty on a fresh install.)

Click **Add New Category** to open the form. Beyond the name and URL key, each category has an optional intro, an **SEO** panel (Meta Title / Description / Keywords) and a **Store Views** selector controlling where the category is visible:

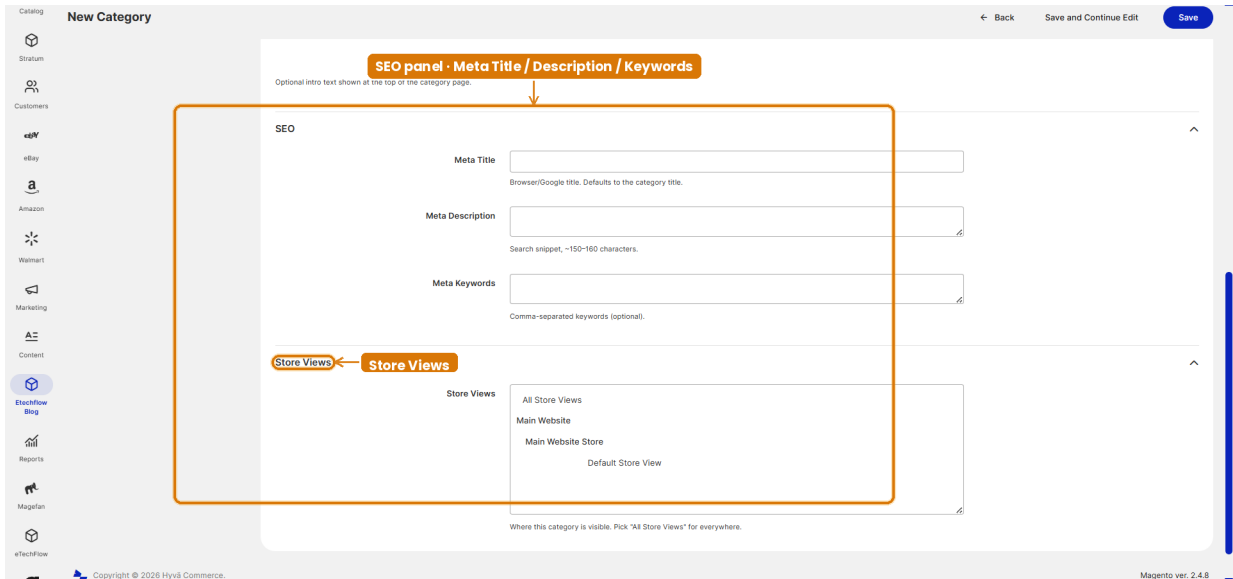


Figure 6 — The New Category form — the SEO panel (Meta Title, Meta Description, Meta Keywords, each with guidance) and the Store Views selector for multi-store visibility.

The category intro accepts rich content through a full editor with layout blocks, headings, buttons and HTML — so a category landing page can be as simple or as designed as you like:

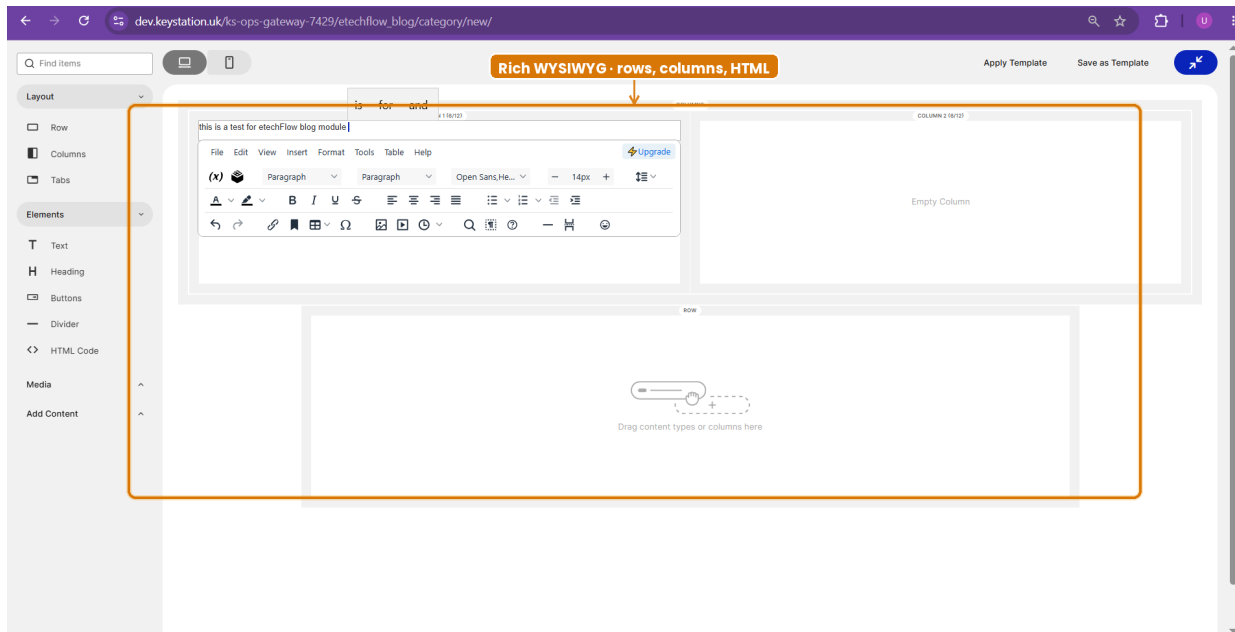


Figure 7 — The category content editor — a rich WYSIWYG / layout builder (rows, columns, text, headings, buttons, HTML) for the optional intro shown at the top of the category page.

## 5.2 Posts

Posts are the articles. Manage them at **Etechflow Blog → Posts**; the grid shows ID, Title, URL Key, Status, **Views** and **Published** date.

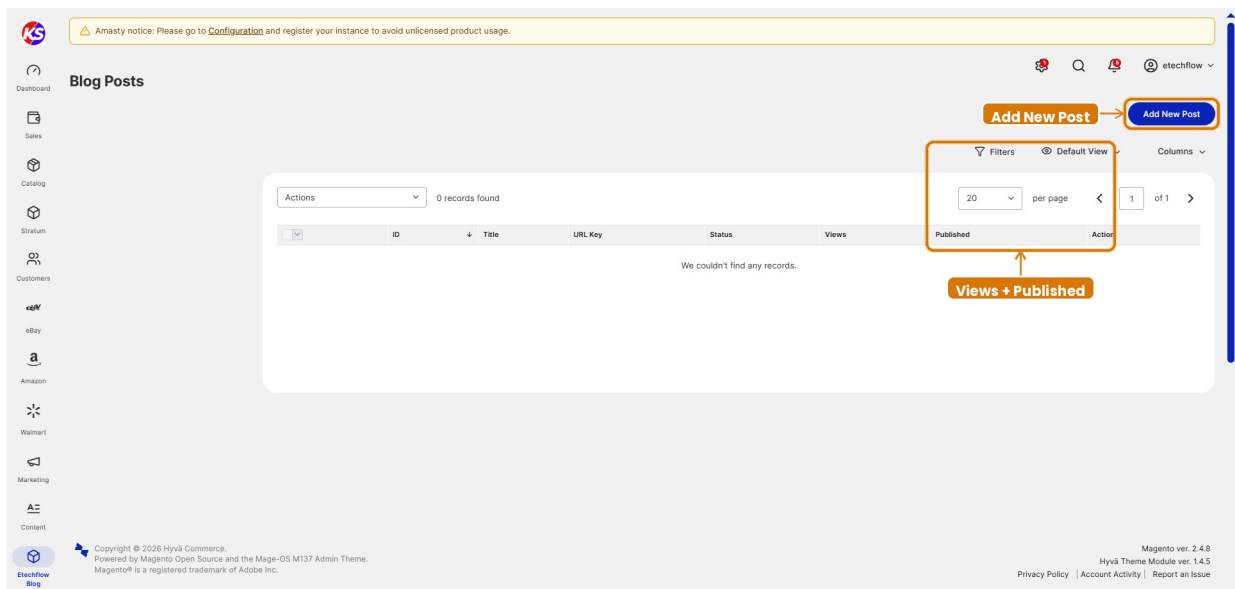


Figure 8 — The Blog Posts grid — including Views and Published columns, with Add New Post, Filters and column controls.

**Add New Post** opens the editor. The **General** panel captures the essentials — an **Enabled** switch (drafts stay hidden), the **Title**, an auto-filled **URL Key**, the **Author**, a **Publish Date** (set a future date to schedule), an optional **Page Heading (H1)**, the **Excerpt**, and the article **Content**:

Figure 9 — The New Post form — General panel: Enabled toggle, Title, URL Key, Author dropdown, Publish Date, Page Heading and Excerpt, with the Content editor below.

### Note

**Author is optional.** A new install has no authors yet, so the Author dropdown shows only “-- Please Select --”; leaving it there saves the post with no author (the author box simply doesn’t render). Create authors under **Etechflow Blog → Authors** to populate the dropdown.

The **Images** panel controls the visuals — a Featured Image (also the social share image), an optional Listing Thumbnail, and responsive hero **Banners** for desktop, tablet and mobile:

Figure 10 — The post Images panel — Featured Image, Listing Thumbnail and Banner (Desktop / Tablet / Mobile), each with fallback behaviour explained inline.

The **Categories & Tags** panel files the post. Pick one or more **Categories** and add free-form **Tags** (each tag creates a `/blog/tag/<tag>` page and feeds the tag cloud). **Include In Recent/Featured** and **Position** control where the post surfaces:

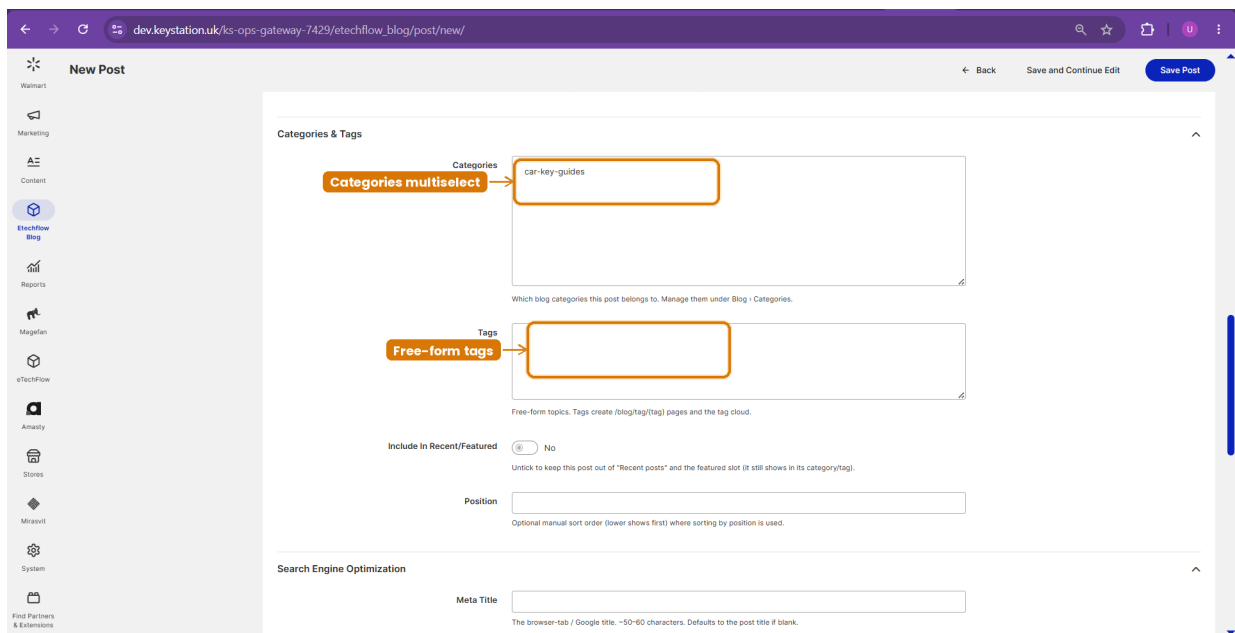


Figure 11 — The post Categories & Tags panel — Categories multiselect, Tags, the Include In Recent/Featured toggle and a manual Position field.

Selecting a category is a single click in the multiselect — here the post is filed under **car-key-guides**:

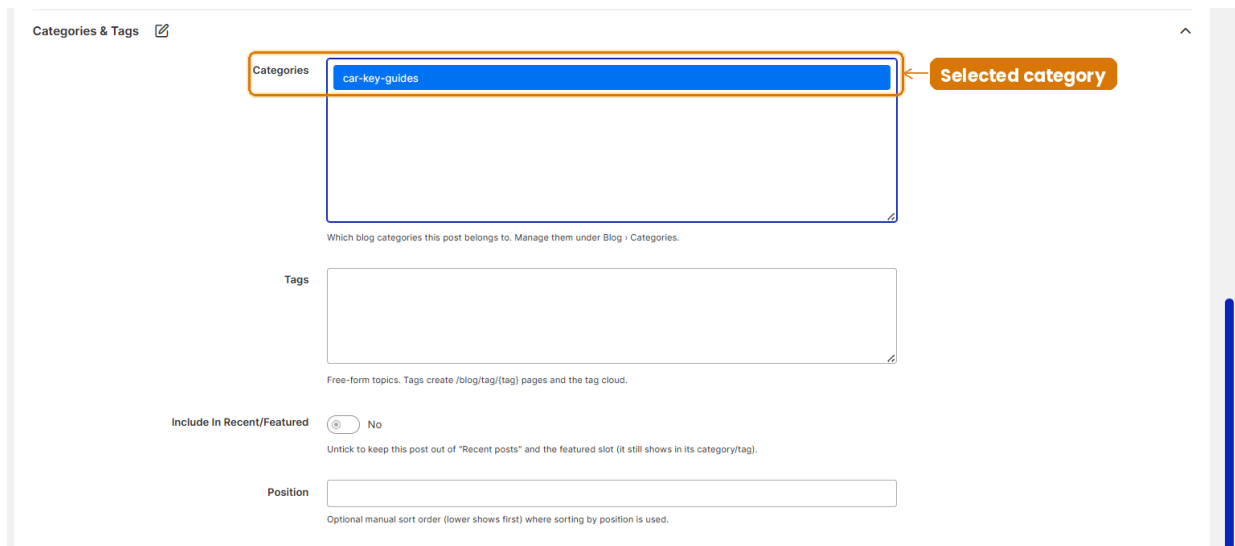


Figure 12 — Assigning a category on the post form — “car-key-guides” selected (highlighted) in the Categories multiselect.

Finally, the **SEO / Open Graph** and **Comments & Store Views** panels control search and social metadata (Meta Robots, Social Title / Description / Image), whether comments are allowed on this specific post, and which store views show it:

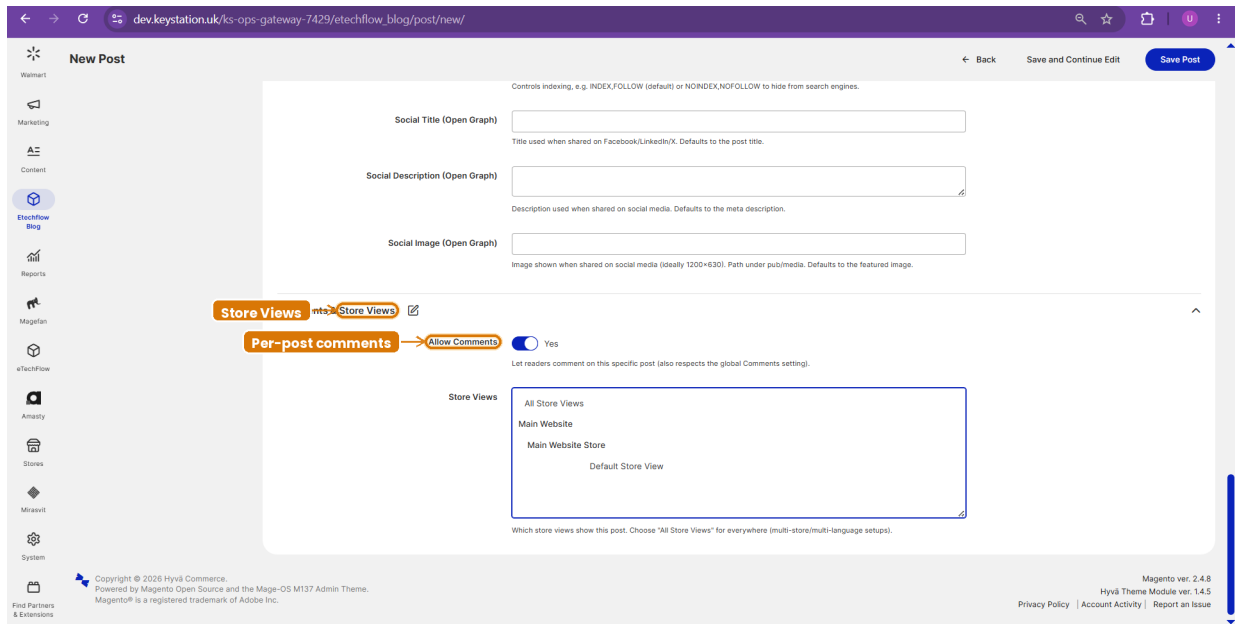


Figure 13 — The post SEO (Open Graph / Twitter) fields plus the Comments & Store Views panel — Allow Comments toggle and the Store Views selector.

## 6 • Comment Moderation

Readers can comment on posts (subject to the global and per-post **Allow Comments** settings). Submissions are held for moderation and never appear publicly until approved — protecting the storefront from spam. Manage them at **Etechflow Blog → Comments**.

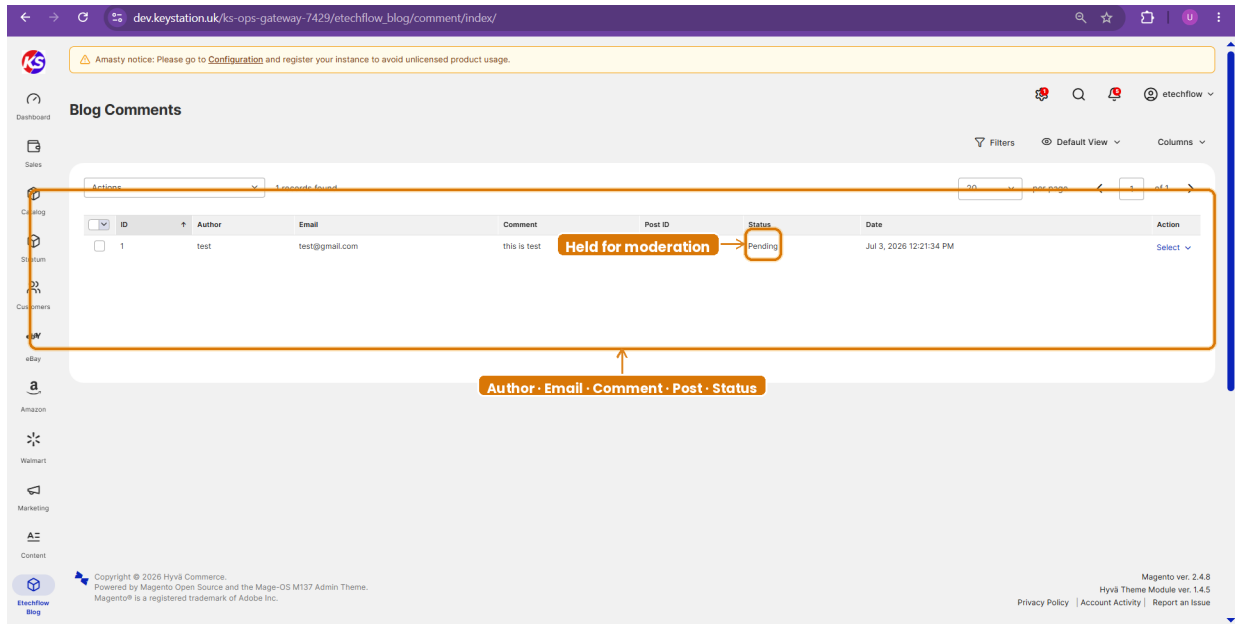


Figure 14 — The Blog Comments moderation grid — ID, Author, Email, Comment, Post ID, Status and Date. A freshly submitted comment shows as Pending.

Select one or more comments and use the **Actions** menu to **Approve**, **Mark Not Approved** or **Delete** them. Only **Approved** comments render on the storefront; Pending and Not-Approved comments stay admin-only.

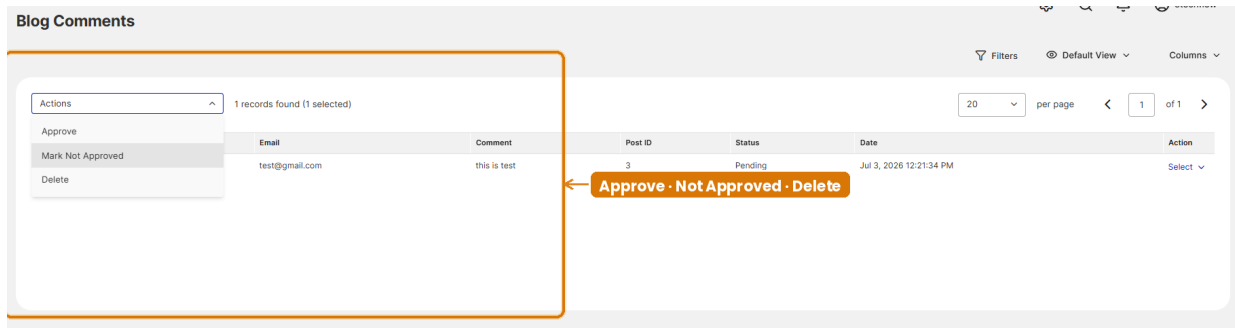


Figure 15 — The comment moderation actions — Approve, Mark Not Approved and Delete — applied to selected comments.

### Important

Comments default to **Pending**, so nothing self-publishes. Reader input is also escaped on output, so a comment containing HTML is shown as literal text rather than executed — the moderation queue plus escaping keep the comment thread safe.

## 7 • Storefront Experience

Shoppers reach the blog at your configured base path (default `/blog`). The blog home lists posts newest-first with a search box and RSS link; each post has its own richly-featured reading page.

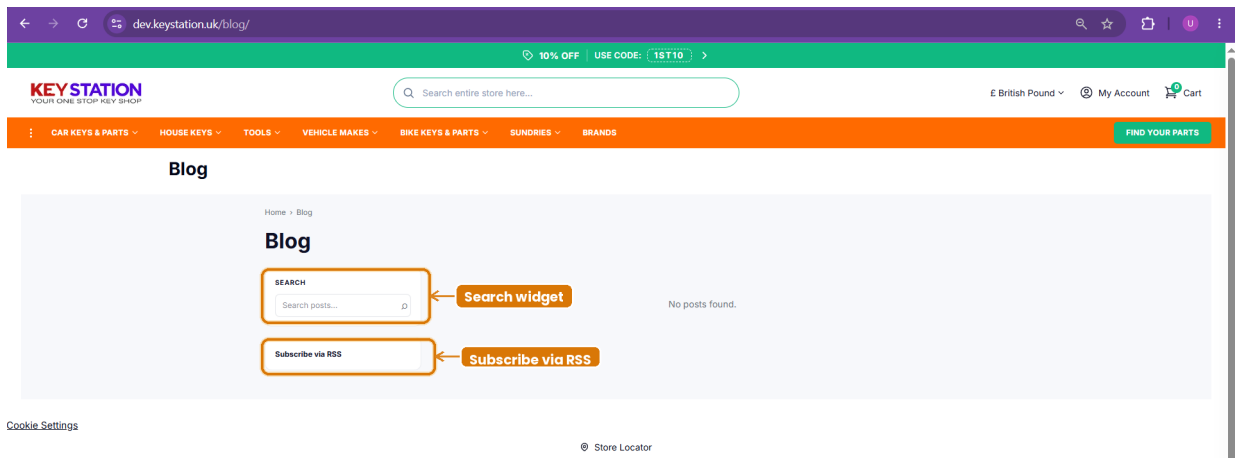


Figure 16 — The storefront blog home — title, breadcrumb, a Search widget and a Subscribe via RSS link. (Shown on a fresh blog before any posts exist.)

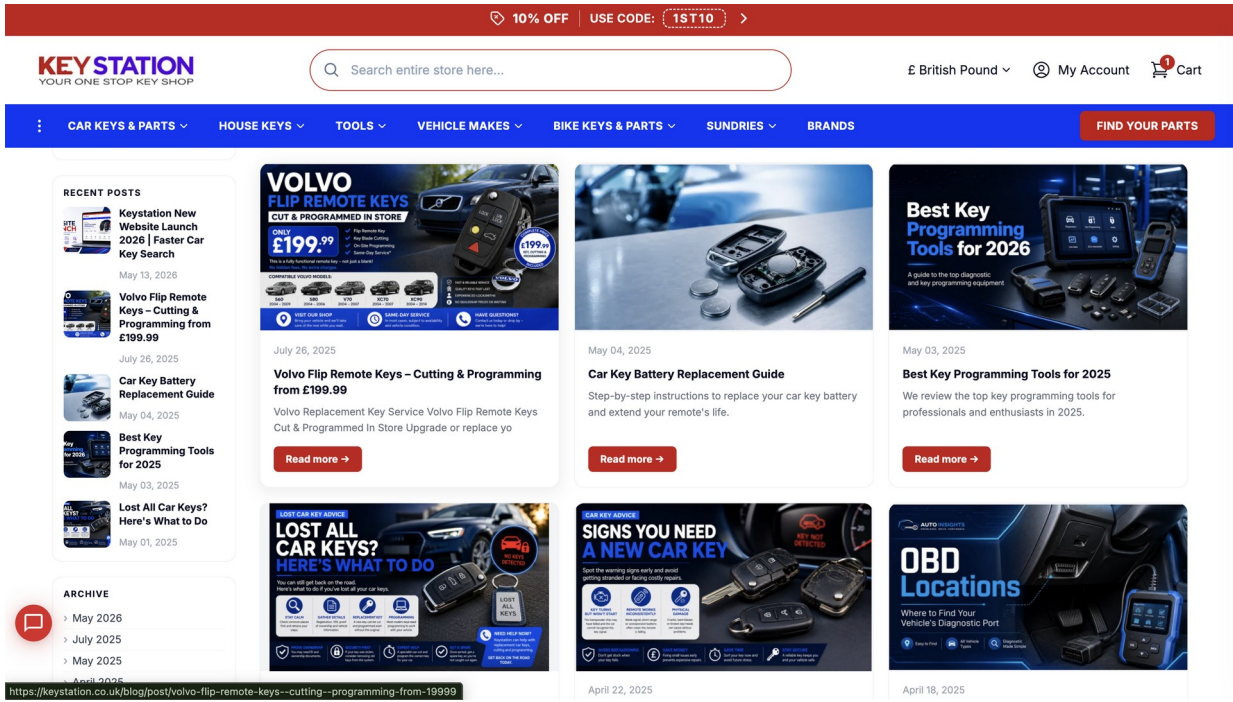


Figure 17 — The same storefront blog home once posts are published — a responsive card grid (image, date, title, excerpt and a Read more link), newest-first, with the Recent Posts and Archive discovery sidebar.

Category pages work the same way, listing only the posts filed under that category:

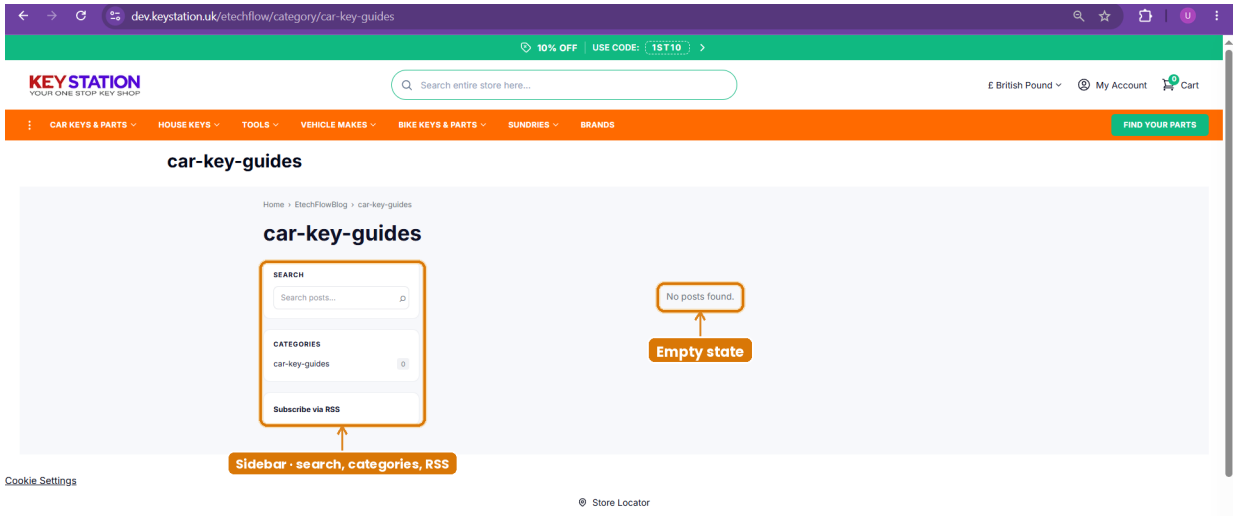


Figure 18 — A category page (“car-key-guides”) — heading, breadcrumb, the sidebar (search, categories, RSS) and the “No posts found” empty state before a published post is assigned.

As soon as content exists, the sidebar fills out with discovery widgets. The **Categories** widget lists each category with its live post count:

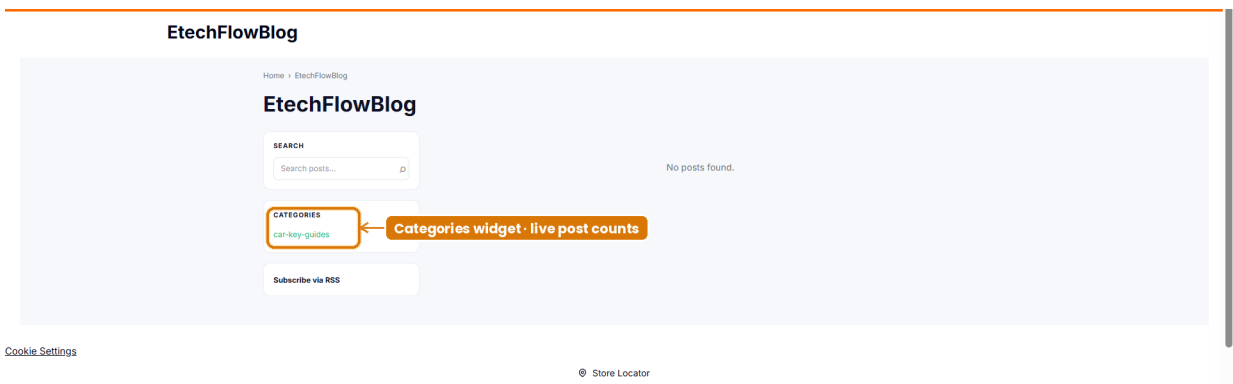


Figure 19 — The blog home sidebar with the Categories widget — each category shown with a post count.

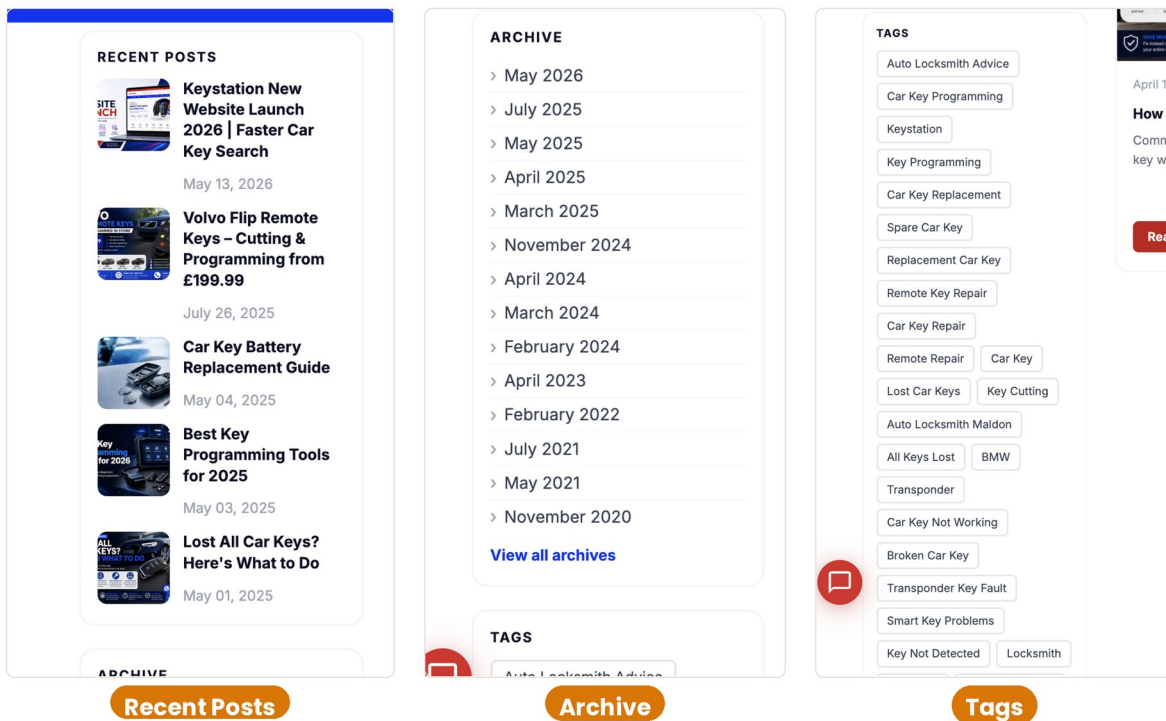


Figure 20 — The discovery sidebar widgets once content exists — Recent Posts (with thumbnails), Archive (by month, with a View all archives link) and a Tags cloud. These sit alongside Search, Categories and the Subscribe via RSS link.

## 7.1 The post page

The single post page is the heart of the reader experience. It shows the title and breadcrumb, the publish date, **reading time** and **views**, **social share** buttons, and a **Comments** section:

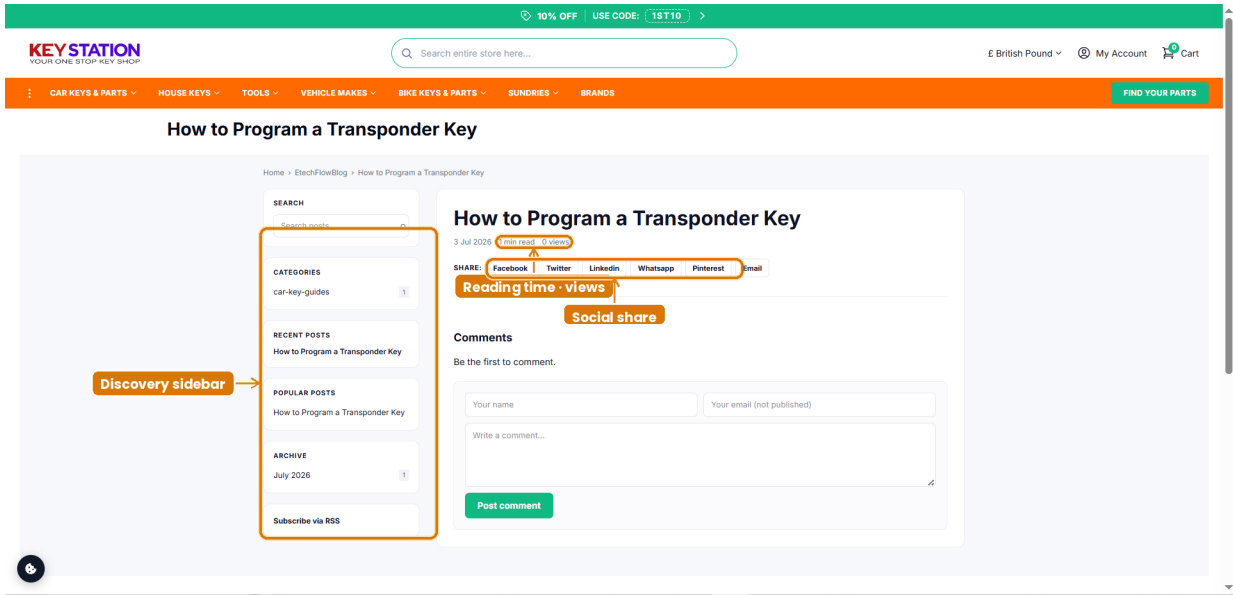


Figure 21 — A single post page — title, breadcrumb, “<date> · X min read · N views”, Facebook/Twitter/LinkedIn/WhatsApp/Pinterest/Email share buttons, a populated sidebar (Search, Categories, Recent, Popular, Archive, RSS) and the comment form.

When the post has body content, the module automatically builds a **table of contents** from its headings and computes an accurate reading time — both derived server-side from the article body:

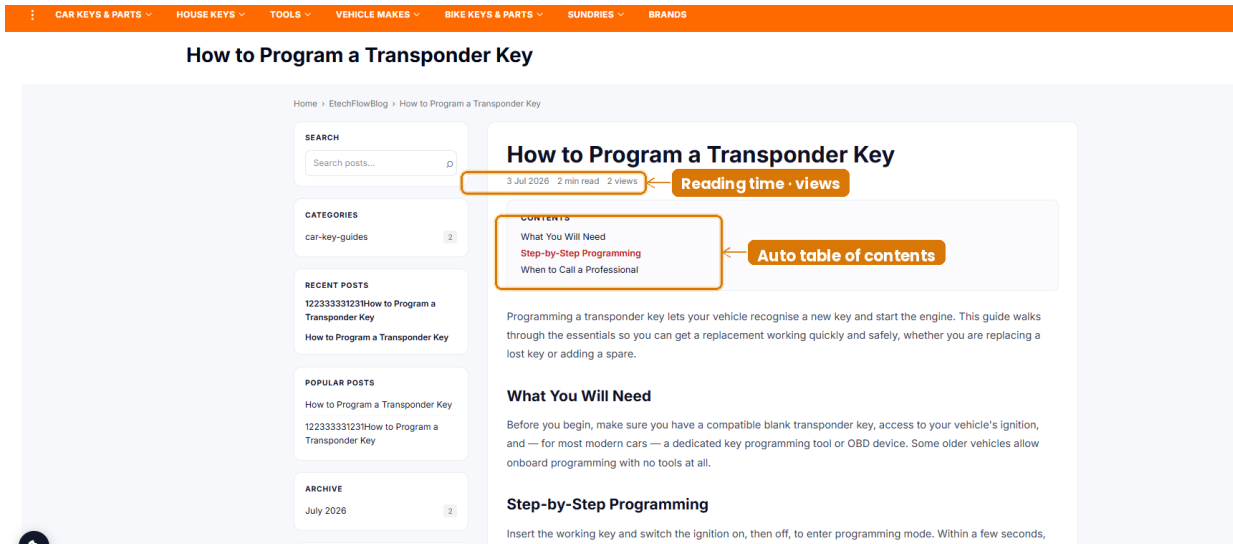


Figure 22 — A post with content — the auto-generated Contents box (built from the H2 headings), “2 min read”, the view counter, and the article body rendered with headings.

Below the article, a “**You may also like**” block surfaces related posts, and approved comments appear above the comment form:

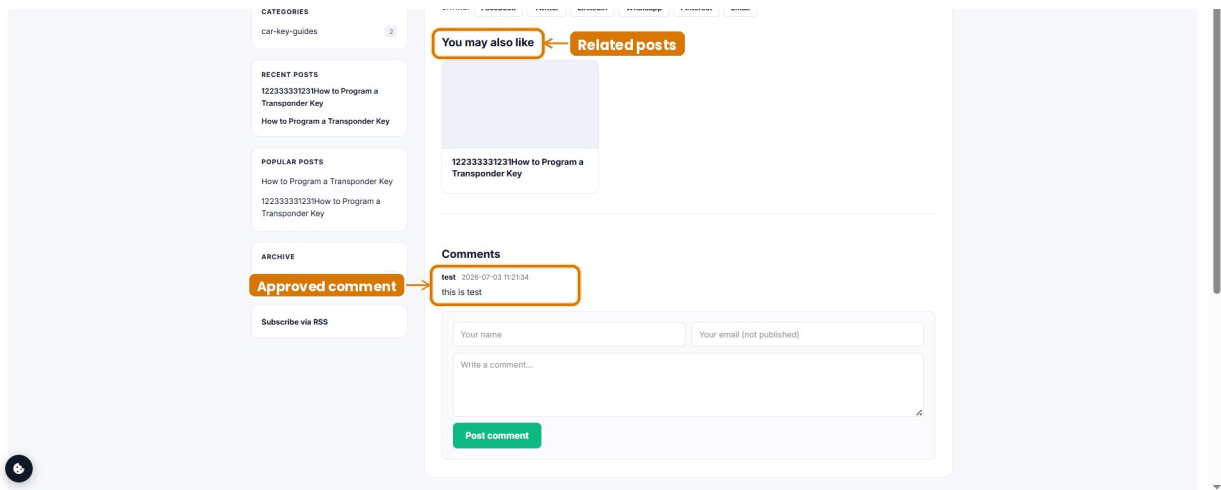


Figure 23 — The lower post page — the “You may also like” related-posts block and an approved comment displayed above the comment form.

## 7.2 Storefront navigation link

The blog is reachable from site navigation. In addition to the optional top-menu link, a **licence-aware footer link** can sit in your footer — it points at the blog’s configured URL and hides automatically if the blog is ever unlicensed:

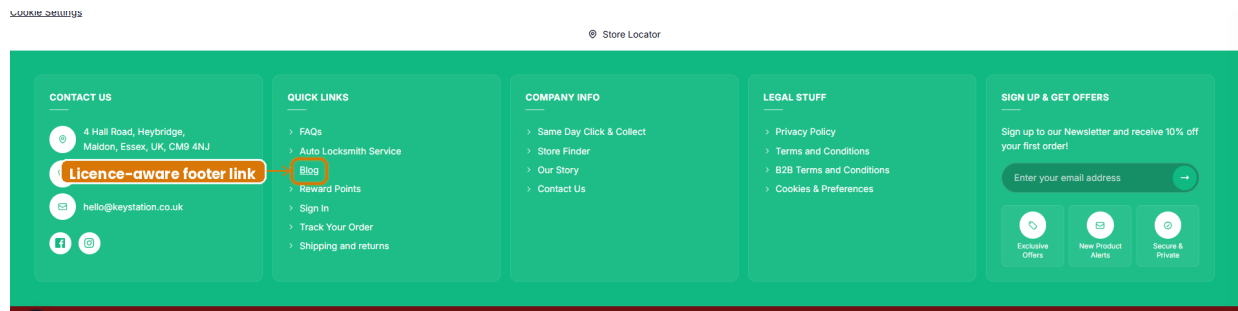


Figure 24 — The storefront footer Quick Links with the Blog link — licence-aware, pointing at the configured blog URL.

## 8 • SEO, Feeds & Discovery

Every blog page is built to be indexed and shared. Each post emits a self-referencing **canonical**, **Open Graph** and **Twitter Card** tags, and **JSON-LD** structured data (BlogPosting, BreadcrumbList, Person, Organization). Viewing the page source confirms the canonical and structured data are present:

```

607 }
608
609 window.alpine.initialized = fn => window.addEventListener('alpine:initialized', fn, {once: true})
610 window.addEventListener('alpine:init', () => Alpine.data('{}', () => {}), {once: true})
611 window.addEventListener('user-allowed-save-cookie', () => InternalCookie.saveTempStorageCookies({})
612
613 })(window.hyva = window.hyva || {});
614 // # sourceURL=hyva.js
615 </script>
616 <script type="speculationrules">
617   "tag": "Hyva Theme",
618   "prerender": {
619     "source": "document",
620     "where": {
621       "and": [
622         "href_matches": "/*",
623         "not": {
624           "href_matches": ["/*customer/*", "/*search/*", "/*sales/*", "/*wishlist/*", "/*checkout/*", "/*paypal/*", "/*newsletter/*", "/*stores/store/redirect/*"]
625         }
626       ]
627     },
628     "selector_matches": [".no-preload", ".do-not-prerender", "[download]", "[rel=nofollow]", "[target]"]
629   }
630 },
631 ],
632 ],
633 ],
634 ],
635 ],
636 ],
637 ],
638 ],
639 ],
640 ],
641 ],
642 ],
643 ],
644 ],
645 ],
646 ],
647 ],
648 ],
649 ],
650 ],
651 ],
652 ],
653 ],
654 ],
655 ],
656 ],
657 ],
658 ],
659 ],
660 ],
661 ],
662 ],
663 ],
664 ],
665 ],
666 ],
667 ],
668 ],
669 ],
670 ],
671 ],
672 ],
673 ],
674 ],
675 ],
676 ],
677 ],
678 ],
679 ],
680 ],
681 ],
682 ],
683 ],
684 ],
685 ],
686 ],
687 ],
688 ],
689 ],
690 ],
691 ],
692 ],
693 ],
694 ],
695 ],
696 ],
697 ],
698 ],
699 ],
700 ],
701 ],
702 ],
703 ],
704 ],
705 ],
706 ],
707 ],
708 ],
709 ],
710 ],
711 ],
712 ],
713 ],
714 ],
715 ],
716 ],
717 ],
718 ],
719 ],
720 ],
721 ],
722 ],
723 ],
724 ],
725 ],
726 ],
727 ],
728 ],
729 ],
730 ],
731 ],
732 ],
733 ],
734 ],
735 ],
736 ],
737 ],
738 ],
739 ],
740 ],
741 ],
742 ],
743 ],
744 ],
745 ],
746 ],
747 ],
748 ],
749 ],
750 ],
751 ],
752 ],
753 ],
754 ],
755 ],
756 ],
757 ],
758 ],
759 ],
760 ],
761 ],
762 ],
763 ],
764 ],
765 ],
766 ],
767 ],
768 ],
769 ],
770 ],
771 ],
772 ],
773 ],
774 ],
775 ],
776 ],
777 ],
778 ],
779 ],
780 ],
781 ],
782 ],
783 ],
784 ],
785 ],
786 ],
787 ],
788 ],
789 ],
790 ],
791 ],
792 ],
793 ],
794 ],
795 ],
796 ],
797 ],
798 ],
799 ],
800 ],
801 ],
802 ],
803 ],
804 ],
805 ],
806 ],
807 ],
808 ],
809 ],
810 ],
811 ],
812 ],
813 ],
814 ],
815 ],
816 ],
817 ],
818 ],
819 ],
820 ],
821 ],
822 ],
823 ],
824 ],
825 ],
826 ],
827 ],
828 ],
829 ],
830 ],
831 ],
832 ],
833 ],
834 ],
835 ],
836 ],
837 ],
838 ],
839 ],
840 ],
841 ],
842 ],
843 ],
844 ],
845 ],
846 ],
847 ],
848 ],
849 ],
850 ],
851 ],
852 ],
853 ],
854 ],
855 ],
856 ],
857 ],
858 ],
859 ],
860 ],
861 ],
862 ],
863 ],
864 ],
865 ],
866 ],
867 ],
868 ],
869 ],
870 ],
871 ],
872 ],
873 ],
874 ],
875 ],
876 ],
877 ],
878 ],
879 ],
880 ],
881 ],
882 ],
883 ],
884 ],
885 ],
886 ],
887 ],
888 ],
889 ],
890 ],
891 ],
892 ],
893 ],
894 ],
895 ],
896 ],
897 ],
898 ],
899 ],
900 ],
901 ],
902 ],
903 ],
904 ],
905 ],
906 ],
907 ],
908 ],
909 ],
910 ],
911 ],
912 ],
913 ],
914 ],
915 ],
916 ],
917 ],
918 ],
919 ],
920 ],
921 ],
922 ],
923 ],
924 ],
925 ],
926 ],
927 ],
928 ],
929 ],
930 ],
931 ],
932 ],
933 ],
934 ],
935 ],
936 ],
937 ],
938 ],
939 ],
940 ],
941 ],
942 ],
943 ],
944 ],
945 ],
946 ],
947 ],
948 ],
949 ],
950 ],
951 ],
952 ],
953 ],
954 ],
955 ],
956 ],
957 ],
958 ],
959 ],
960 ],
961 ],
962 ],
963 ],
964 ],
965 ],
966 ],
967 ],
968 ],
969 ],
970 ],
971 ],
972 ],
973 ],
974 ],
975 ],
976 ],
977 ],
978 ],
979 ],
980 ],
981 ],
982 ],
983 ],
984 ],
985 ],
986 ],
987 ],
988 ],
989 ],
990 ],
991 ],
992 ],
993 ],
994 ],
995 ],
996 ],
997 ],
998 ],
999 ],
1000 ],

```

Figure 25 — The rendered post page source — the self-referencing canonical link (highlighted) alongside the JSON-LD structured-data block.

### 8.1 RSS feed

The blog exposes a valid **RSS 2.0** feed at `/blog/rss` (it follows the configured base path, e.g. `/etechflow/rss`). The channel uses your blog title and base URL, and each item carries the post title, link, GUID, publish date and an auto-generated description — so readers and aggregators can subscribe. A **Subscribe via RSS** link appears in the storefront sidebar.

## 9 • Product-Page Integration

The blog integrates with the catalog: a post can be linked to the products it discusses, and those products then show a **“Related articles”** block on their product pages — turning helpful content into a path back to the catalog.

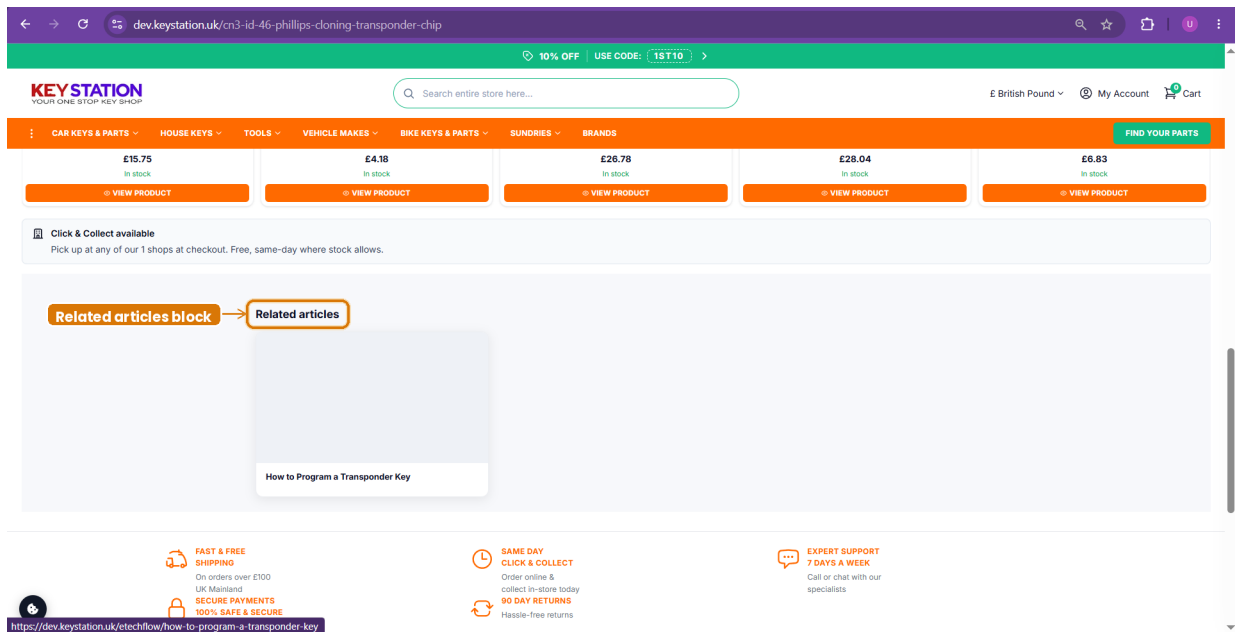


Figure 26 — A product page (“CN3 ID:46 Phillips Cloning Transponder Chip”) with the blog’s “Related articles” block, surfacing the linked post “How to Program a Transponder Key”.

You curate the link from the **post** side. On the post edit form, the **Related Products** panel provides an **Add Related Products** button:

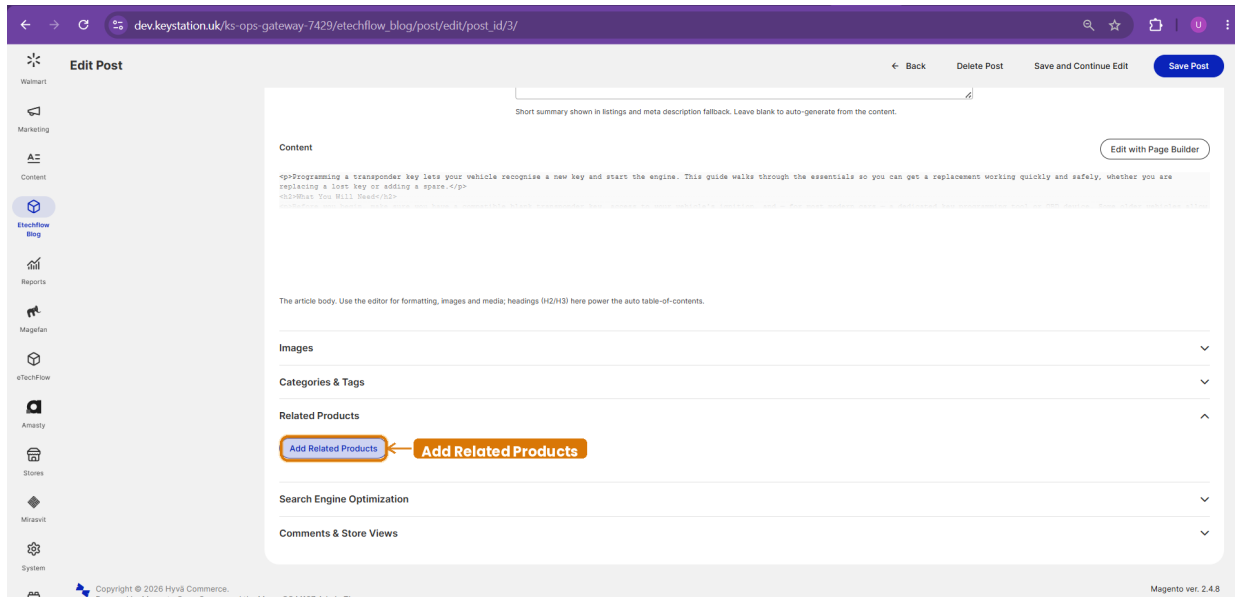


Figure 27 — The post edit form — the Related Products panel with the Add Related Products button, alongside the article Content and the other collapsible panels.

The button opens a searchable, filterable **product grid** in a modal. Tick the products this post relates to, confirm, and save — each selected product’s page then shows the post under “Related articles”:

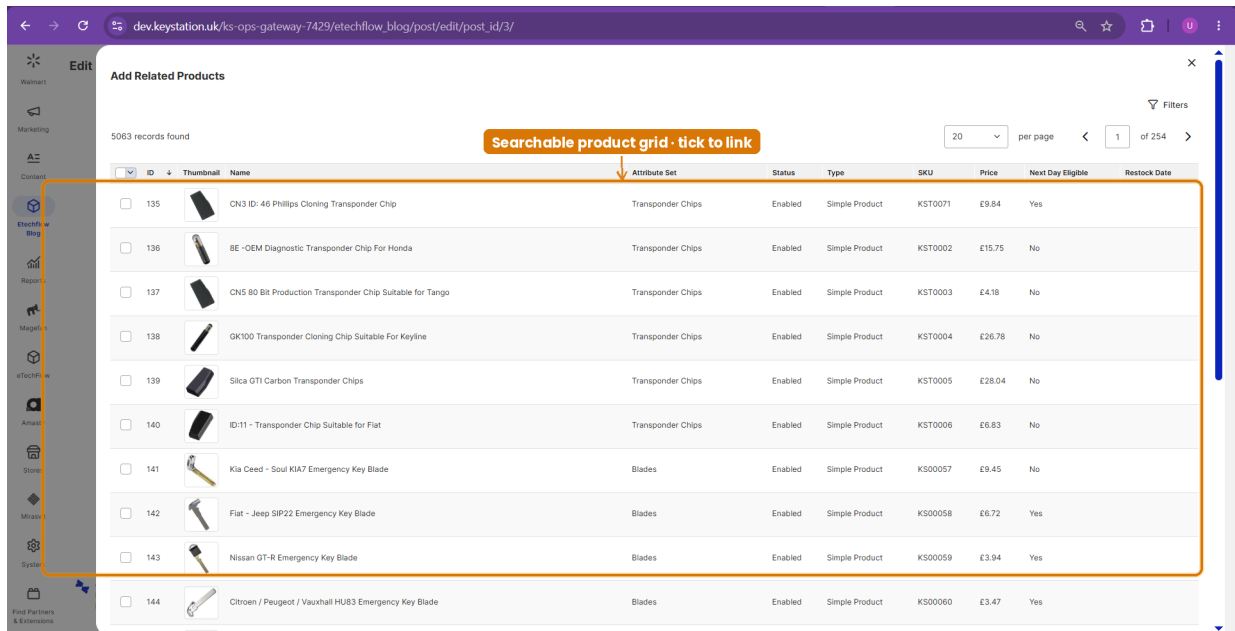


Figure 28 — The Add Related Products modal — a full product grid (thumbnail, ID, SKU, price, stock) with checkboxes for selecting the products a post relates to.

## 10 • Configuration Reference

Stores → Configuration → Etechflow → Blog (etechflow\_blog). The key General options:

Setting (path)	Default	Purpose
etechflow_blog/general/enabled	Yes	Master on/off switch. No hides the whole blog (all /blog URLs become unavailable).
etechflow_blog/general/route	blog	First segment of every blog URL, e.g. “blog” gives /blog, /blog/my-post. Flush cache after changing.
etechflow_blog/general/title	Blog	The H1 / page title on the blog home, and the breadcrumb label.
etechflow_blog/general/posts_per_page	9	Posts per listing page before pagination (recommended 9–12).
etechflow_blog/general/show_short_content	Yes	Show an excerpt preview of each post in listings (auto-generated if no manual excerpt).
etechflow_blog/general/short_content_limit	200	Maximum length of an auto-generated excerpt (characters).
etechflow_blog/general/show_in_top_menu	Yes	Add a link to the blog in the storefront main navigation menu.
etechflow_blog/general/top_menu_text	Blog	The wording of the top-menu link (e.g. Blog, News, Guides).
etechflow_blog/comments/moderation	Yes	Hold submitted comments for approval before they appear on the storefront.
License Portal URL	license-service.../license/validate	Endpoint the module validates the licence against. Change only if your portal URL changes.

## 11 • Licensing & Activation

The blog is activated with a key from the eTechFlow portal, validated live against `license-service.etchflow.com` and bound to your licensed domain.

1. Obtain a key (SP-XXXX-XXXX-XXXX-XXXX) from your eTechFlow account — it is also your Composer download token.
2. Go to **Etechflow Blog → License** (or Stores → Configuration → Etechflow → Blog → License), paste the key and Save.
3. Flush the cache. The blog validates the key within its short cache window and unlocks.

### Important

Full-block behaviour: when the licence is missing, suspended, expired or bound to a different domain, the whole blog route returns 404, the blog admin redirects to the licence gate, and the storefront top-menu / footer links and the product-page “Related articles” block all hide. Development / staging domains (hosts starting with dev., staging., qa., ...) run unlocked automatically; an optional “Enforce Licence on This Domain” toggle lets you validate a production licence on a staging URL before go-live.

## 12 • Troubleshooting / FAQ

**I changed a setting but the storefront didn't update.**

On a production store, saving configuration invalidates the Full Page Cache but does not rebuild it. Flush the cache (and hard-refresh your browser). The Blog Base URL Path field's own hint reminds you of this.

### **The blog pages look unstyled.**

Static content was deployed for the wrong locale. In production run `setup:static-content:deploy` for your store's actual locale (e.g. `en_GB`), then flush the cache — otherwise the browser can't load the blog's CSS.

### **A category shows “No posts found”.**

A post appears in a category only when it is BOTH Enabled (not a draft) AND assigned to that category on the post form. Check both, then flush the cache.

### **I can't save a post — an author error appears.**

Author is optional — leave it on “-- Please Select --” and the post saves with no author. If you want an author, create one under Etechflow Blog → Authors first, then pick it on the post.

### **The Table of Contents / reading time doesn't show.**

Both are computed from the post body. The Contents box only appears when the article has headings (H2/H3); an empty post has nothing to summarise.

### **A comment I approved isn't on the post.**

Only Approved comments render on the storefront, and the post page is cached — flush the cache after approving. Pending and Not-Approved comments never appear publicly.

### **How do I link a product to a post?**

On the post edit form, open the Related Products panel, click Add Related Products, tick the products in the modal grid and Save. Those products' pages then show the post under “Related articles”.

### **/blog returns a 404.**

Either the blog is disabled, or the licence is missing / suspended / expired / wrong-domain — the blog is fully blocked (404) until a valid licence is in place.