

Store Locator

[Store & branch finder](#) • [Leaflet + OpenStreetMap map](#) • [postcode proximity search](#) • [no paid map API key](#)

An admin-managed store / branch finder for Magento 2 / Adobe Commerce. Shoppers find their nearest store on an interactive Leaflet + OpenStreetMap map and by postcode proximity search (powered by the free postcodes.io service). You manage every store — address, phone, opening hours, coordinates — from a simple admin grid. Turn-by-turn directions and Street View are one tap away, and there is no paid map API key to buy.

Etechflow_StoreLocator • **Module Documentation & User Guide**

This guide is written for Hyvä • Magento / Adobe Commerce 2.4.6 – 2.4.9 • PHP 8.1 – 8.4 • Part of the eTechFlow Suite

© eTechFlow • 

Table of Contents

1. Overview
2. Features at a Glance
3. Architecture & How It Works
4. Installation & Setup
5. Managing Stores (Admin)
6. Configuration Reference
7. The Storefront Finder
8. Licensing & Activation
9. Troubleshooting / FAQ

1 • Overview

1.1 What the module does

Store Locator gives your storefront a professional **store / branch finder**: an interactive map of all your locations and a **postcode proximity search** that lists the shopper's nearest stores by distance. Each store is managed from the Magento admin — name, address, phone, email, opening hours, a tagline and its map coordinates — and appears on the storefront with **Get Directions**, **Street View** and **View Store** actions. The map uses **free Leaflet + OpenStreetMap** tiles and the postcode lookup uses the free **postcodes.io** service, so there is **no paid map API key** to purchase.

This guide is for Hyvä

This documentation — and every screenshot in it — is for the module running on a **Hyvä** storefront theme. The screenshots come from a live Hyvä store (██████████), running the **Hyvä Theme Module 1.4.5** on **Magento 2.4.8**. The storefront finder is built with Hyvä + Alpine.js, and the admin uses the Hyvä-compatible Mage-OS admin theme. (The module also runs on Luma, but this guide targets Hyvä.)

1.2 Highlights

- **Interactive map** — Leaflet + OpenStreetMap, with a marker for every active store. **No Google Maps billing / API key required.**
- **Postcode proximity search** — the shopper enters a UK postcode; the finder lists the nearest stores, sorted by distance, within an adjustable radius.
- **Admin store manager** — a full grid + form to add, edit and delete stores (address, phone, email, opening hours, tagline, coordinates, sort order).
- **Turn-by-turn directions & Street View** — one-tap deep links to Google Maps directions and Street View for each store (no API key — just links).
- **View Store links** — send shoppers from a store card to a landing page you choose (e.g. /stores/london).
- **Auto footer link** — a licence-aware “Store Locator” link is added to the storefront so shoppers can always find the page.
- **SEO-friendly canonical URL** — the finder lives at a single clean URL (/store-locator); common variants (/store-finder, /stores) redirect to it.
- **Fully responsive** — the hero, search bar, map and store cards adapt to mobile and desktop.
- **Licence-gated** — part of the eTechFlow Suite; activated with an eTechFlow licence key, validated against your domain.

1.3 Compatibility matrix

Area	Supported
Magento / Adobe Commerce	2.4.6 – 2.4.9 (Open Source & Commerce)
PHP	8.1 – 8.4
Storefront theme	Hyvä (this guide) — also Luma-compatible
Map	Leaflet + OpenStreetMap (no paid API key)
Geocoding / proximity	postcodes.io (UK postcodes), distance computed in-browser
Licensing	eTechFlow ██████████ (SP key / HMAC / Bundle key)

2 • Features at a Glance

Feature	What it does
Interactive map	Leaflet + OpenStreetMap map with a pin for each active store; click a pin for a popup.
Postcode proximity search	Geocodes the shopper's postcode (postcodes.io) and lists nearest stores within the radius, sorted by distance.
Adjustable radius	A slider (in miles) filters the results live; its starting value is the admin Default Search Radius.
Store cards	Each store shows name, address, phone, tagline, opening hours, distance, and action buttons.
Get Directions	Opens Google Maps turn-by-turn directions to the store's coordinates.
Street View	Opens Google Street View at the store's coordinates.
View Store	Links to a store landing page (the admin Store URL).
Admin store manager	Grid + form to create, edit, delete and sort stores.
Footer link	A licence-aware "Store Locator" link injected into the storefront footer.
Brand copy	The finder's intro copy uses your Brand Name (e.g. "Visit your local <Brand> store").

3 • Architecture & How It Works

3.1 The module

A self-contained module — **Etechflow_StoreLocator** (package `etechflow/module-store-locator`). It ships an admin **store manager** (grid + form) backed by a single `etechflow_store_locator` table, a storefront **finder page**, a small AJAX endpoint that returns the store data, the Leaflet/OpenStreetMap front-end, and the eTechFlow licence layer. Its configuration lives under the **Etechflow** tab, labelled **Store Locator**.

3.2 How it works

- **Stores** are records in `etechflow_store_locator` — each with an address and a **latitude / longitude** used to place its map pin.
- The finder page (route `store-finder`, served at the canonical URL `/store-locator`) renders the Leaflet + OpenStreetMap map and the search UI.
- A lightweight endpoint (`store-finder/stores/search`) returns **all active stores** as JSON — coordinates included.
- **Proximity is computed in the browser**: the shopper's postcode is geocoded via **postcodes.io**, then a haversine distance to each store is calculated, filtered by the radius and sorted nearest-first.
- **Directions & Street View** are deep links to Google Maps — no Google API key is used; the map itself is OpenStreetMap.
- A **licence gate** (`LicenseValidator::isValid()`) protects the storefront finder and the admin store manager; without a valid licence the module is blocked.

3.3 Built with Hyvä

On a Hyvä theme the finder is a Hyvä + **Alpine.js** component (`x-data`, `x-for`, `x-model`): the map, the live radius slider, the postcode search and the store-card list are all reactive. The admin screens use Magento UI components on the Hyvä-compatible admin theme.

4 • Installation & Setup

4.1 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.etcchflow composer
[redacted] /composer

composer config http-basic.[redacted] you@example.com SP-XXXX-XXXX-XXXX-XXXX

composer require etcchflow/module-store-locator

bin/magento module:enable Etechflow_StoreLocator

bin/magento setup:upgrade

bin/magento setup:di:compile          # production mode

bin/magento setup:static-content:deploy # production mode (ships map JS/CSS)

bin/magento cache:flush
```

Production mode note

Because the finder ships front-end assets (the Leaflet map JS/CSS), a store in **production** mode must run **setup:di:compile** and **setup:static-content:deploy** after enabling — otherwise the map assets won't be served. In **developer** mode, **setup:upgrade** + **cache:flush** is enough.

4.2 First-run steps

1. Paste your licence key under **Stores → Configuration → Etechflow → Store Locator → License** (see Section 8).
2. Set your **General Settings** — Brand Name and Default Search Radius (Section 6).
3. Add your stores under **Stores → Store Locator → Manage Stores, including latitude & longitude** (Section 5).
4. Visit `/store-locator`` on the storefront to see the finder, and confirm the footer link appears.

5 • Managing Stores (Admin)

5.1 Where to find it

The module adds a **Store Locator** group to the admin **Stores** menu, with two entries — **License** and **Manage Stores**:

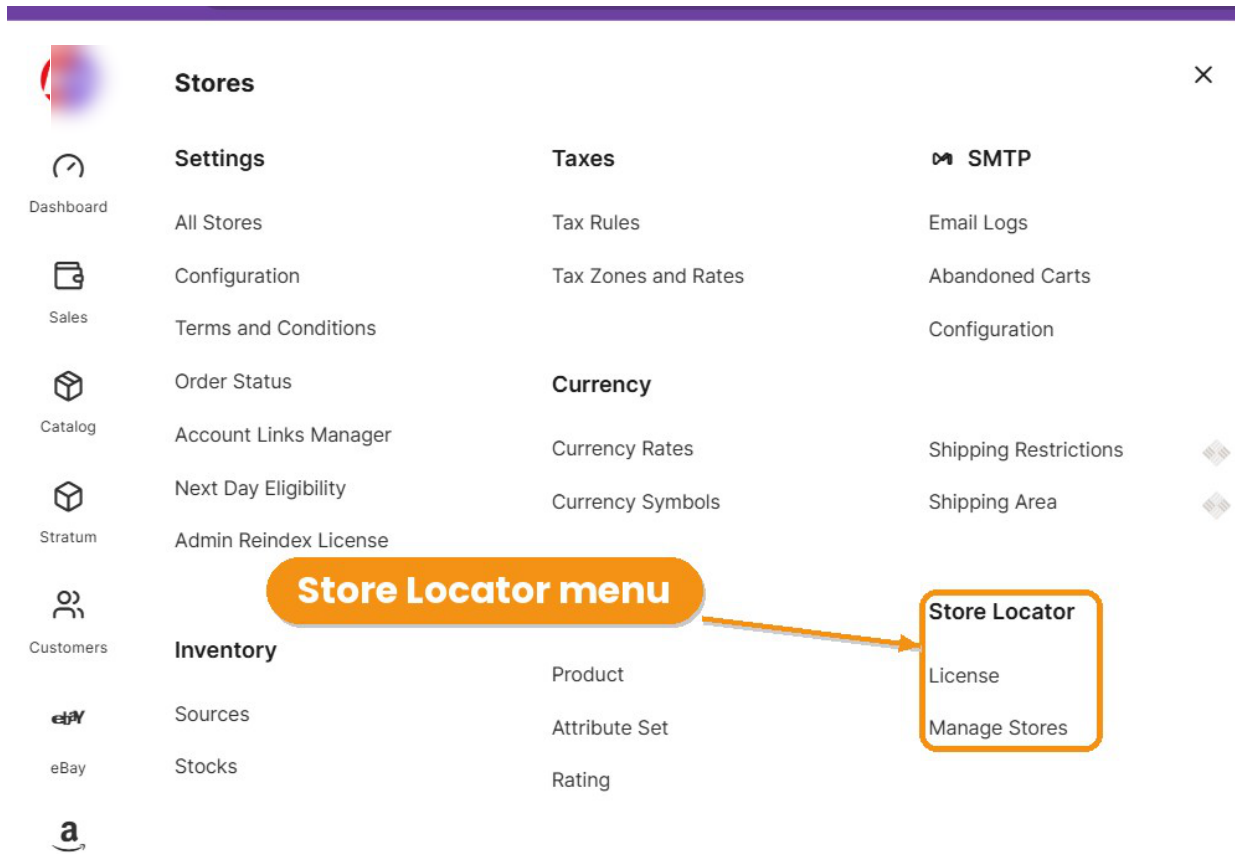


Figure 1 — the admin Stores menu — the Store Locator group with its License and Manage Stores entries.

5.2 The store grid

Stores → Store Locator → Manage Stores lists every store with searchable, filterable, sortable columns (ID, Store Name, City, Postcode, Phone, Status) and an **Add New Store** button. Use **Actions → Edit** or **Delete** on any row.

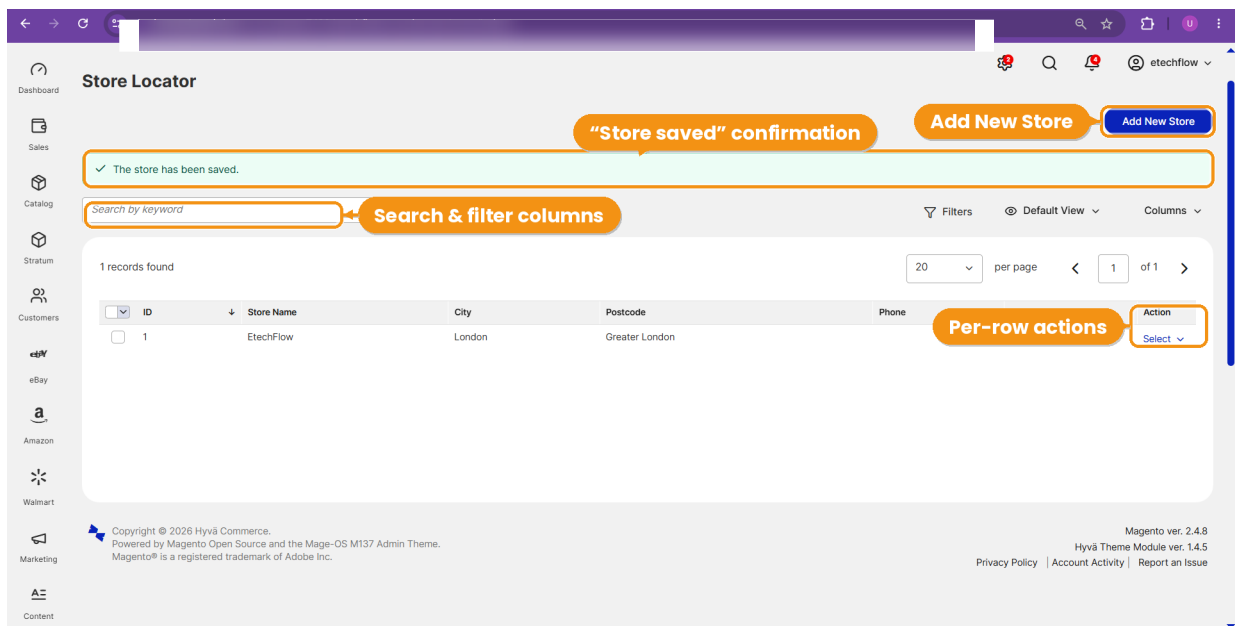


Figure 2 — the Manage Stores grid after saving a store — searchable columns, per-row actions, and the “store has been saved” confirmation.

5.3 Adding / editing a store

The store form captures everything the storefront card and map need. The first group covers identity and address:

The screenshot shows the 'New Store' form in the Magento admin interface. The form is titled 'New Store' and has a 'Save Store' button. It contains several input fields: 'Active' (a toggle switch set to 'Yes'), 'Store Name' (required, with 'EtechFlow' entered), 'Street' (123 Oxford St), 'City' (London), 'County' (Greater London), 'Postcode' (Greater London), 'Phone', 'Email', and 'Taxline'. Orange callouts highlight the 'Active: publish store' toggle, the 'Store Name' field, and a group of fields including 'Street', 'City', 'County', and 'Postcode' labeled 'Address · city · postcode'.

Figure 3 — the New Store form (top) — Active toggle, Store Name (required), Street, City, County, Postcode, Phone, Email.

...and the second half covers the tagline, opening hours, the all-important **coordinates**, the View-Store link and sort order:

The screenshot shows the lower half of the 'New Store' form. It contains input fields for 'Email', 'Tagline' (with a short strapline shown under the store name), 'Opening Hours' (with an example: Mon-Fri 9:00am-5:00pm), 'Latitude' (51.5074, with an example: 51.7400), 'Longitude' (-0.1278, with an example: -0.1276), 'Store URL' (stores/london, with a note: Relative path for View Store button (e.g. /stores/my-store)), and 'Sort Order' (10). Orange callouts highlight the 'Opening hours' field, a group of fields including 'Latitude' and 'Longitude' labeled 'Map coordinates (lat / long)', and the 'Store URL · View Store link' field.

Figure 4 — the New Store form (lower half) — Tagline, Opening Hours, Latitude, Longitude, Store URL and Sort Order.

The full field list:

Field	Purpose
Active	Toggle. Only active stores appear on the storefront. Default On.
Store Name	Required. Shown as the card title and map popup heading.
Street / City / County / Postcode	The store's address, shown on the card.
Phone / Email	Contact details shown on the card.
Tagline	A short strapline shown under the store name.
Opening Hours	Free text, e.g. "Mon-Fri 9:00am-5:00pm".
Latitude / Longitude	The map coordinates — used for the pin AND the proximity search. See the note below.
Store URL	Relative path for the "View Store" button, e.g. /stores/london.
Sort Order	Controls the order stores are listed in.

Enter latitude & longitude for every store

The store's **coordinates** — not its postcode — place its map pin and drive the proximity search. Enter **Latitude** and **Longitude** for each store (e.g. London is 51.5074, -0.1278). A store saved without coordinates won't appear on the map or be found by a radius search. You can copy coordinates from Google Maps (right-click a location → the lat/long is at the top).

6 • Configuration Reference

Stores → Configuration → Etechflow → Store Locator (etechflow_storelocator). The section opens with a **Module Status** banner (licence + module state), then the License and General Settings groups:

The screenshot shows the 'Store Locator' configuration page. On the left is a sidebar with various system modules. The main content area is titled 'Configuration' and contains two sections: 'License' and 'General Settings'.
 In the 'License' section, there are two fields: 'License Portal URL' (with a placeholder and a note to update if the portal URL changes) and 'Bundle License Key' (with a placeholder and a note that it's a shared key for the module).
 In the 'General Settings' section, there are three fields: 'Brand Name' (with a placeholder and a note to leave blank for the configured store name), 'Google Maps API Key' (with a placeholder and a note that it's optional and uses free Leaflet + OpenStreetMap), and 'Default Search Radius (km)' (with a placeholder and a note that it's the default radius in kilometres for the store finder search).
 Annotations on the right side of the 'General Settings' section point to the 'Brand Name', 'Google Maps API Key', and 'Default Search Radius (km)' fields with labels: 'Brand Name - storefront copy', 'Optional - leave blank', and 'Default search radius' respectively.

Figure 5 — the Store Locator configuration — the License group (Portal URL, Bundle Key) and General Settings (Brand Name, Google Maps API Key, Default Search Radius).

6.1 General Settings

Setting	Default	Purpose
---------	---------	---------

Brand Name	(blank)	Used in the storefront copy, e.g. “Visit your local <Brand> store”. Blank → falls back to your Store Name.
Google Maps API Key	(blank)	Optional and not required — the map uses free Leaflet + OpenStreetMap. Leave blank.
Default Search Radius (miles)	10	The starting value of the storefront radius slider. Distances on the storefront are shown in miles.

No paid map key needed

The **Google Maps API Key** field is optional and can be left blank — the map is powered by free **Leaflet + OpenStreetMap** tiles, and directions/Street View are simple deep links. There is nothing to buy and no billing to configure.

6.2 License

Setting	Purpose
Module Status	A banner showing whether the module is active for this domain (see Section 8).
License Key	Your SP-XXXX subscription key or a per-module HMAC key. Bound to your domain.
Enforce Licence on This Domain	Optional. Requires a valid licence even on development/staging domains (see Section 8).
License Portal URL	The eTechFlow endpoint that validates keys — default <code>/license/validate</code> .
Bundle License Key	One shared key that activates every eTechFlow module on the host (stored encrypted).

7 • The Storefront Finder

The finder lives at `/store-locator``. It opens with a hero, then a search bar (postcode / address + a radius slider), the interactive map, and a list of store cards:

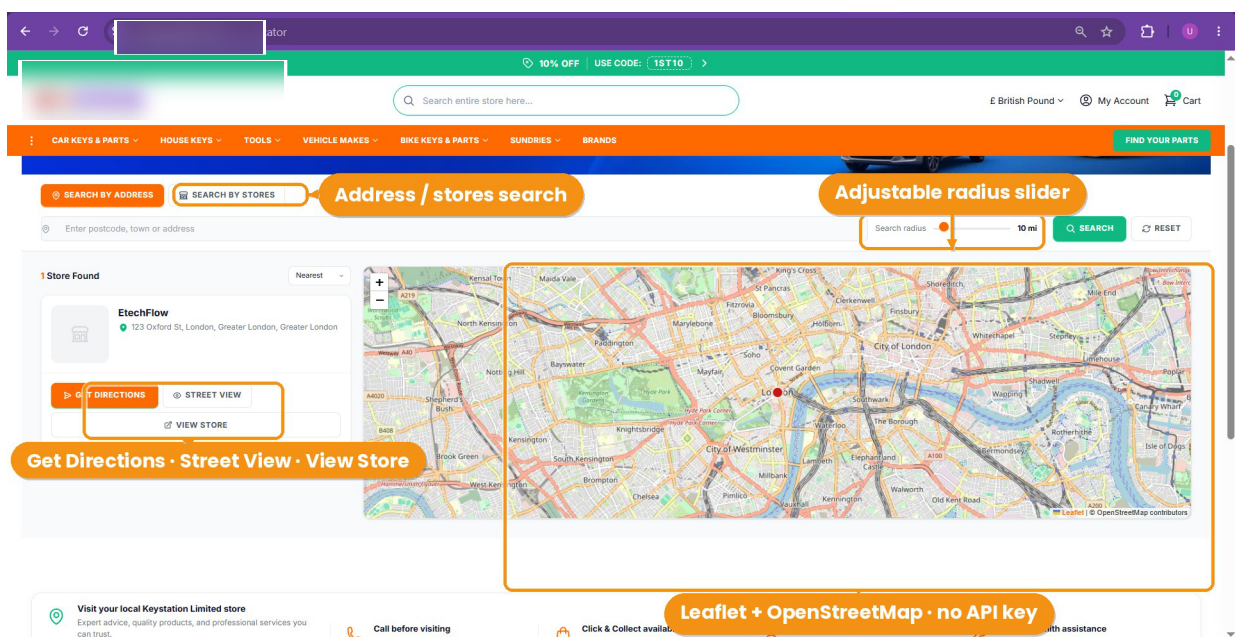


Figure 6 — the storefront finder on Hyvä — Search by Address / Search by Stores, the radius slider, the Leaflet + OpenStreetMap map, and the store card with Get Directions / Street View / View Store.

7.1 Postcode proximity search

The shopper types a **UK postcode** (or town/address) and sets the **radius**. The finder geocodes the postcode via **postcodes.io**, calculates the distance to each store, and shows the nearest stores — each card displaying its **distance** (in miles) — filtered by the radius and sorted nearest-first. Dragging the radius slider re-filters the list live.

- **Widen the radius** to see more stores; **narrow it** to see only the closest.
- An out-of-range search shows a friendly “no stores found — try expanding your search radius” message.
- The proximity search is designed for **UK postcodes** (postcodes.io). The map and store list work everywhere; the postcode lookup is UK-specific.

7.2 Map, directions & Street View

Each active store is a pin on the map. Clicking a pin (or a store card) zooms to that store and opens its popup. Each card carries three actions:

Action	What it does
Get Directions	Opens Google Maps turn-by-turn directions to the store's coordinates in a new tab.
Street View	Opens Google Street View at the store's coordinates.
View Store	Opens the store's landing page (the Store URL you set in the admin).

7.3 Footer link & entry points

So shoppers can always reach the finder, the module adds a licence-aware **Store Locator** link to the storefront (footer / trust strip), such as the “Find your nearest store” call-to-action:

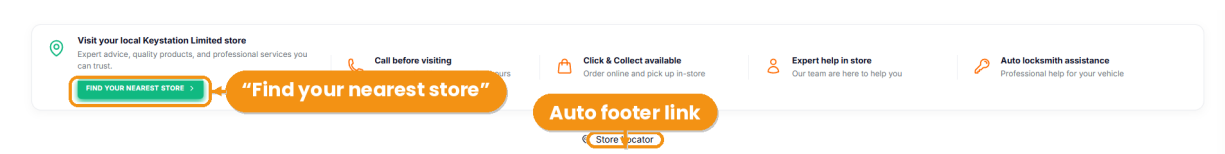


Figure 7 — the storefront trust strip with the “Find your nearest store” button and the “Store Locator” footer link.

The finder also has a stable, SEO-friendly URL. It is served at the canonical ``/store-locator``, and common variants such as ``/store-finder`` and ``/stores`` redirect to it, so there is only ever one indexable store-finder URL.

8 • Licensing & Activation

Store Locator is part of the **eTechFlow Suite**, activated with a key from the eTechFlow portal and validated against , bound to your licensed domain. Without a valid licence the storefront finder and the admin store manager are blocked.

5. Obtain a key — an **SP-XXXX-XXXX-XXXX-XXXX** subscription key, or a shared **Bundle** key that activates every eTechFlow module on the host. The key is also your Composer download token.
6. Enter it under **Store Locator** → **License** → **License Key** (or set the suite-wide **Bundle License Key**), then **Save Config**.

7. The **Module Status** banner at the top of the section shows the current state — a green “active — licensed for <domain>” chip when the licence is valid.

8.1 The Module Status banner

Banner	Meaning
Green — “active, licensed for <domain>”	A genuine licence has validated for this domain. The full module is unlocked.
Blue — “active (development domain)”	The site is on a development / staging domain (e.g. dev. *, staging. *), so enforcement is bypassed and the module runs unlocked with no key. A real licence will be needed on your production domain.
Amber — “inactive”	No valid licence (missing, wrong domain, suspended or expired) — the storefront finder and admin store manager are locked.

Enforce Licence on This Domain

Development and staging domains normally run unlocked so agencies don't burn a licence while building. If you want to validate your **real production licence on a dev/staging URL before go-live**, set **License → Enforce Licence on This Domain = Yes**; the module will then require and validate a real key even on that domain.

9 • Troubleshooting / FAQ

The map is a grey box.

The map must render in a visible container. The current version waits for the map container to have a real size before initialising, so this is handled — if you still see grey, hard-refresh the page and make sure the finder isn't inside a tab/accordion that starts hidden.

My stores don't appear on the map / aren't found by the postcode search.

Each store needs **Latitude and Longitude**. A store saved with a blank lat/lng won't get a map pin and can't be distance-matched. Edit the store and add its coordinates (copy them from Google Maps).

The postcode search finds nothing.

Check the postcode is a valid **UK** postcode (the proximity lookup uses postcodes.io, which is UK-only), and try **widening the radius**. A far-away postcode with a small radius correctly returns no stores.

Is the radius in miles or kilometres?

The storefront finder works in **miles** — the radius slider and the distances on the cards are miles, and the admin **Default Search Radius** is in miles too.

The Default Search Radius I set isn't used on the storefront.

Update to the latest version — the storefront now starts the radius slider at the admin **Default Search Radius** value, then the shopper can adjust it.

The banner says “development domain” — is the module licensed?

On a development/staging domain (e.g. dev.*) the module runs unlocked without a key by design. To require a real licence there, set **License → Enforce Licence on This Domain = Yes** and enter your key. On your production domain a valid licence is always required.

Do I need a Google Maps API key?

No. The map is free Leaflet + OpenStreetMap, and Get Directions / Street View are simple Google Maps deep links. The Google Maps API Key field is optional — leave it blank.

Does it work on Luma?

This guide is written for **Hyvä**, and the screenshots are from a Hyvä store. The module is also Luma-compatible, but for the experience shown here, run it on a Hyvä theme.

I set up the finder but shoppers can't find the page.

The module adds a licence-aware **Store Locator** link to the storefront footer, and the finder is at the canonical `/store-locator` (with `/store-finder` and `/stores` redirecting to it). You can also link to it from your own menu or a CMS block.