

# Rich Snippets (JSON-LD)

[Hyvä-native structured data](#) · [schema.org JSON-LD](#) · [Product](#) · [Breadcrumb](#) · [ItemList](#) · [Organization](#) · [WebSite](#) · [OpenGraph + Twitter](#)

A Hyvä-native structured-data extension for Magento 2 / Adobe Commerce. It emits clean schema.org JSON-LD — Product (price, availability, brand, MPN/GTIN, condition, reviews), Breadcrumbs, category ItemLists, your Organization and WebSite (with a Sitelinks search box), and WebPage — plus OpenGraph and Twitter Card meta for social sharing. Everything is toggle-controlled per snippet type, with admin-mappable attribute codes so it fits any catalog.

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

Built for Hyvä themes · 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. What Structured Data It Outputs
3. Architecture & How It Works
4. Installation & Setup
5. Configuration Reference
6. Verifying the Snippets
7. Licensing & Activation
8. Troubleshooting / FAQ

## 1 • Overview

### 1.1 What the module does

**Rich Snippets (JSON-LD)** adds **structured data** to your storefront so search engines can show **rich results** — star ratings, price, availability, breadcrumbs, a sitelinks search box — and so links shared on social media render a proper card. It outputs standards-compliant **schema.org JSON-LD** on your product, category, CMS and home pages, plus **OpenGraph** and **Twitter Card** meta for social sharing. Each snippet type is individually switchable, and the product attributes it reads (brand, MPN, GTIN, condition) are mapped in admin so it fits any catalog.

#### Built for Hyvä

This is a **Hyvä-native** module. Its output is generated through Hyvä ViewModels and layout, and it is designed to run on **Hyvä storefront themes**. The screenshots in this guide are from a live Hyvä store (Keystation).

### 1.2 Highlights

- **schema.org JSON-LD** — clean, valid `<script type="application/ld+json">` output, not fragile inline microdata.
- **Product** rich results — name, price, currency, availability, condition, brand, MPN/GTIN, and reviews (AggregateRating).
- **ProductGroup** for configurable products — every variant listed via `hasVariant`.
- **Category ItemList + BreadcrumbList** — helps Google map your catalog structure.
- **Organization + WebSite** with a **Sitelinks Search Box** (SearchAction) — sitewide.
- **WebPage** schema on the home page and CMS pages, linked to your Organization/WebSite.
- **OpenGraph + Twitter Card** meta — rich link previews on Facebook, WhatsApp, X, LinkedIn.
- **Per-type toggles + attribute mapping** — turn each snippet on/off; point brand/MPN/GTIN/condition at your own attribute codes.
- **Coexistence-aware** — a master switch keeps it OFF until you're ready, so it never doubles up with another SEO extension.

### 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 (Hyvä-native)                                     |
| Output                   | schema.org JSON-LD + OpenGraph + Twitter Card meta     |
| Licensing                | eTechFlow license-service (SP key / HMAC / Bundle key) |

## 2 • What Structured Data It Outputs

With the module enabled, each page type carries the relevant schema.org JSON-LD (and, on products, OpenGraph/Twitter meta). This is the full picture:

| Page               | JSON-LD types emitted  | Purpose  |
|--------------------|--|--|
| Product            | Product · Offer/AggregateOffer · AggregateRating + Review · Brand · ProductGroup (configurables) | Price, availability, condition, brand, MPN/GTIN, ratings → rich product results. |
| Category / listing | ItemList · BreadcrumbList  | Lists the products on the page and the category trail.                           |

|                |   |  |
|----------------|---|--|
| Home & CMS     | WebPage · BreadcrumbList                                  | Marks these as real pages within a defined site.   |
| Every page     | Organization · WebSite (+ SearchAction)                   | Your business identity and a Sitelinks Search Box. |
| Product (meta) | OpenGraph (og:type=product, product:price) · Twitter Card | Rich link previews when the product is shared.     |

The **Product** block is the headline feature — here it is in a live Hyvä page's source, with name, sku, image and the schema.org context:

```

0039 </div>
0040 </div>
0041 </div><script type="application/ld+json">{"@context":"https://schema.org", "@type":"Product", "name":"Xhorse (VVDI) & Keydiy Remote Key Fixing Pins \u2013 X 10 for Securing Blade to P-Type Remotes", "sku":"KS02400", "image":"https://
0042 @media(max-width:374px){
0043 .ks-tb-section .container { padding-left:.5rem;padding-right:.5rem;padding-top:1rem;padding-bottom:1rem }
0044 .ks-tb-section .grid { grid-template-columns:repeat(3,1fr) !important;
0045 .ks-tb-section .flex.items-start { flex-direction:column !important;
0046 .ks-tb-section svg { width:24px !important;height:24px !important;flex
0047 .ks-tb-section p { font-size:18px !important;line-height:1.2 !important;text-align:center }
0048 .ks-tb-section .text-gray-500 { display:none !important }
0049 }
0050 @media(max-width:374px){
0051 .ks-tb-section .grid { gap:.7rem .25rem !important }
0052 .ks-tb-section svg { width:20px !important;height:20px !important }
0053 .ks-tb-section p { font-size:9px !important }
0054 }

```

Figure 1 — the Product JSON-LD in a Hyvä product page's source (@type Product, name, sku, image).

Inside it, the **offer** carries price, currency, stock status, condition and a validity date:

The screenshot shows a browser's developer tools source view. A search box at the top right contains the word "condition". Below the search bar, a snippet of JSON-LD is visible, with a callout box pointing to the "offer" object. The callout box contains the text "Offer: price · availability · condition". The JSON-LD snippet includes the following fields for the offer: "price": "4.73", "priceCurrency": "GBP", "availability": "https://schema.org/InStock", "itemCondition": "https://schema.org/NewCondition", "priceValidUntil": "2027-07-02", "sku": "KS0071", and "seller": {"@id": "3-1d-46-phillips-cloning-transponder-chip"}. The browser's address bar shows the URL: "view-source:https://dev.keystation.uk/cn3-id-46-phillips-cloning-transponder-chip".

Figure 2 — the offer: price + priceCurrency (GBP), availability = InStock, itemCondition = NewCondition, priceValidUntil, sku, seller.

### 3 • Architecture & How It Works

#### 3.1 The module

A self-contained module — **Etechflow\_RichSnippets** (package etechflow/module-rich-snippets). It ships Hyvä **ViewModels** (Product, Category, CmsPage, Site, Breadcrumbs, OpenGraph) that build the JSON-LD, layout handles that inject them on the right page types, an admin configuration section, and the eTechFlow licence layer. Its config lives under the **Etechflow** tab, labelled **Rich Snippets (JSON-LD)**.

#### 3.2 How it works

- **Hyvä ViewModels** assemble each schema.org graph from the current product / category / CMS page and your store settings.

- **Layout handles** (catalog\_product\_view, catalog\_category\_view, cms\_page\_view, cms\_index\_index, default) place the JSON-LD <script> blocks and the OpenGraph/Twitter meta.
- **Attribute mapping** — brand / MPN / GTIN-13 / condition are read from the attribute codes you set, so the same module fits any catalog's attribute naming.
- **Licence gate** — everything is behind `LicenseValidator::isValid()`; without a valid licence (and the master switch on) the module emits nothing.

### 3.3 Where it runs — Hyvä

The output is produced through Hyvä-native ViewModels; the surrounding markup you'll see in source is Hyvä / Alpine.js (x-data, x-for). The screenshots throughout are from a Hyvä store:

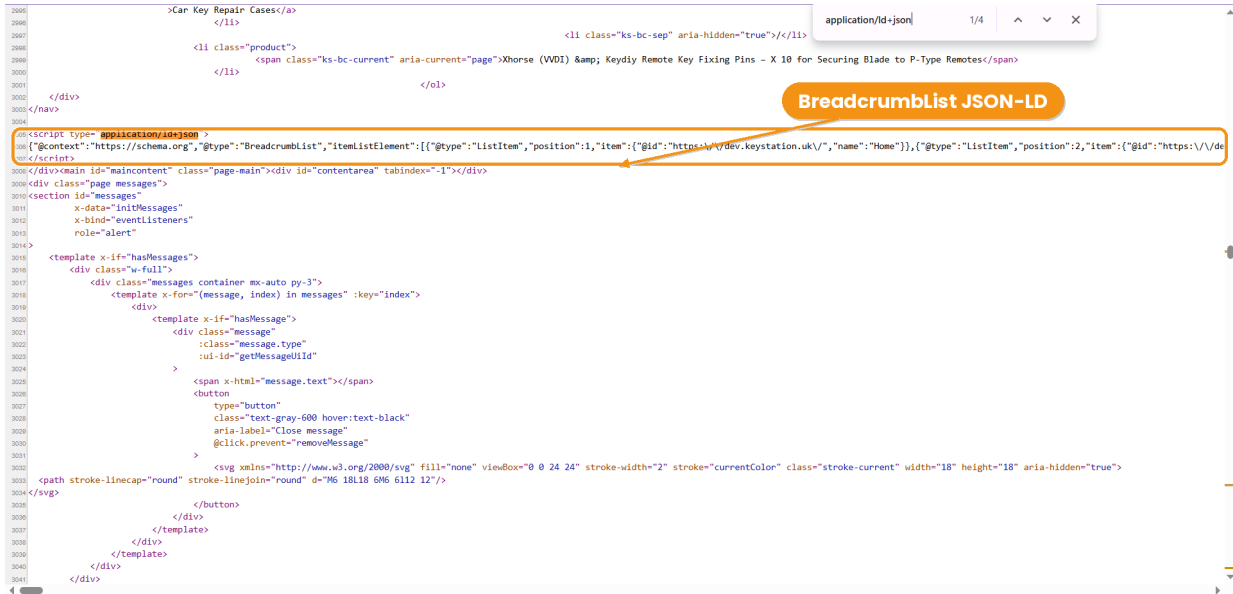


Figure 3 — a Hyvä page's source — the module's BreadcrumbList JSON-LD sits alongside Hyvä/Alpine markup (x-data, x-for).

## 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.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-rich-snippets

bin/magento module:enable Etechflow_RichSnippets

bin/magento setup:upgrade

bin/magento setup:di:compile      # production mode

bin/magento cache:flush
```

## 4.2 First-run steps

1. Paste your licence key under **Stores → Configuration → Etechflow → Rich Snippets → License** (see Section 7).
2. Turn on the **master switch**: General → \*Enable module\* = **Yes**.
3. Leave the per-type toggles (Product / Category / OpenGraph / CMS) at their defaults, or switch off any you don't want.
4. Map the product attribute codes (Brand / MPN / GTIN-13 / condition) to your store's attributes — see Section 5.

### Coexistence

If you already run another SEO/structured-data extension (e.g. Mirasvit), keep this module's master switch **OFF** until you've disabled the other one's structured data. Running both emits **duplicate** JSON-LD / OpenGraph, which confuses search engines and social scrapers. Pick one source.

## 5 • Configuration Reference

**Stores → Configuration → Etechflow → Rich Snippets (JSON-LD)** (etechflow\_richsnippets). The section opens with a **Module Status** banner (licence + module state), then the groups below.

General

Enable module (master switch) [store view] Yes ? **Master switch: off = no schema**  
Master switch. While OFF, the module emits no schema. A valid licence is also required.

Product page

Enable product rich snippet [store view] Yes ? **Enable product rich snippet**

Emit ProductGroup for configurables [store view] Yes ?  
Outputs ProductGroup + hasVariant for configurable products.

Brand attribute code [store view] ?  
Default: brand

MPN attribute code [store view] ?  
Default: sku

GTIN-13 attribute code [store view] ?  
Leave empty if products have no barcode.

Item condition attribute code [store view] ?  
Optional. Values containing used/refurbished/damaged map to schema.org conditions; otherwise NewCondition.

**Attribute mapping: brand · MPN · GTIN · condition**

Figure 4 — the General master switch and the Product page group (enable, ProductGroup, and the Brand / MPN / GTIN-13 / condition attribute codes).

### 5.1 General

| Setting                       | Default | Purpose   |
|-------------------------------|---------|---|
| Enable module (master switch) | No      | Master on/off. While OFF the module emits no schema (a valid licence is also required). |

## 5.2 Product page

| Setting                             | Default | Purpose  |
|-------------------------------------|---------|--|
| Enable product rich snippet         | Yes     | Emit the Product JSON-LD (price, availability, brand, reviews).  |
| Emit ProductGroup for configurables | Yes     | Output ProductGroup + hasVariant for configurable products (no effect on simple products).                     |
| Brand attribute code                | brand   | Product attribute that holds the brand → brand.name. Point it at your real code (e.g. manufacturer).           |
| MPN attribute code                  | sku     | Attribute for the Manufacturer Part Number → mpn. Many stores use sku.   |
| GTIN-13 attribute code              | (blank) | Attribute holding the 13-digit barcode (EAN) → gtin13. Leave blank for products without a barcode.             |
| Item condition attribute code       | (blank) | Optional. Values containing used / refurbished / damaged map to schema.org conditions; otherwise NewCondition. |

## 5.3 Category page

| Setting                      | Default | Purpose  |
|------------------------------|---------|--|
| Enable category rich snippet | Yes     | Emit an ItemList of the products on the page, plus a BreadcrumbList. |

## 5.4 OpenGraph & Twitter

| Setting                         | Default | Purpose  |
|---------------------------------|---------|--|
| Enable OpenGraph + Twitter meta | Yes     | Add OpenGraph + Twitter Card tags so shared links show a title, image and price. |

## 5.5 CMS & home pages

| Setting                            | Default | Purpose  |
|------------------------------------|---------|--|
| Enable WebPage schema (CMS + home) | Yes     | Add WebPage structured data to CMS pages and the home page, linked to your Organization and WebSite. |

## 5.6 License

| Setting            | Purpose  |
|--------------------|--|
| License Key        | Your SP-XXXX subscription key, a per-module HMAC key, or a shared bundle key. Bound to your domain.  |
| Bundle License Key | One shared key that activates every eTechFlow module on the host (stored encrypted).                 |
| License Portal URL | The eTechFlow endpoint that validates keys — default license-service.etechflow.com/license/validate. |
| Last Issued Key    | Read-only audit field, auto-managed by the module.   |

## 6 · Verifying the Snippets

Because a pre-launch store is usually behind a password, Google's live **Rich Results Test** can't fetch it — so verify from the page source and paste the JSON into a validator:

5. Open the page → right-click → **View Page Source** → **Ctrl+F** for `application/ld+json` (or `og: / twitter:`).
6. Copy the JSON block → paste into [validator.schema.org](https://validator.schema.org) (Schema Markup Validator) → it reports the type and any errors.

### 6.1 Product

The Product block should carry name, sku, image, an offers node (price / currency / availability / itemCondition), plus brand, mpn and gtin13 when those attributes are populated. Here the **MPN** resolves from the SKU (the default):

```
ing Transponder Chip "sku":"KST0071", "mpn": "KST0071", "image": "https://res.cloudinary.com/dhzhg1xd/image/upload/c_pad,dpr_2.0,f_auto,q_auto/v1/media/catalog/r  
q": [], "hyva_commerce_messages": [{"messages": [{"mpn resolves from SKU"}]}], "recently_compared_product": [], "product_data_storage": [], "paypal-billing-agreement": [], "paypal-buyer-country": []}</script>
```

Figure 5 — mpn resolved from the product's SKU (KST0071) — the default MPN mapping.

### 6.2 OpenGraph & Twitter (product)

On a product, the social meta is product-specific — `og:type=product`, `product:price:*`, and a large Twitter card with the product's title and image:

```
6 <meta property="product:price:currency" content="GBP" />  
7 <meta property="product:availability" content="in stock" />  
8 <meta name="twitter:card" content="summary_large_image" />  
9 <meta name="twitter:title" content="Cloning&#x20;Transponder&#x20;Chip&#x20;Key&#x20;Station" />  
10 <meta name="twitter:description" content="Bill&#x20;and&#x20;replacement&#x20;keys&#x20;with&#x20;Cloning&#x20;Chip&#x20;Quality&#x20;immobiliser&#x20;chip&#x20;for&#x20;  
11 <meta name="twitter:image" content="https&#x3a;8&#x2f;8&#x2f;res.cloudinary.com&#x2f;dhzhg1xd&#x2f;image&#x2f;upload&#x2f;c_pad,dpr_2.0,f_auto,h_800,q_auto,v_000&#x2f;v1&#x2f;media&#x2f;catalog&#x2f;product&#x2f;c&#x2f;n8&#x2f;cn  
12 <script>
```

Figure 6 — product OpenGraph + Twitter Card: `twitter:card = summary_large_image`, `product:price:currency = GBP`, `product:availability = in stock`.

### 6.3 Category & home

Category pages carry a `BreadcrumbList` (and `ItemList`); the home page and CMS pages carry a `WebPage` block.



### Note

A single **Bundle** key is the simplest option if you run several eTechFlow modules — set it once and every module on that host is licensed. Use a per-module SP key to license only this module.

## 8 • Troubleshooting / FAQ

### Nothing shows in the source — no JSON-LD.

Check two things: the **master switch** (General → Enable module = Yes) and a **valid licence** (Module Status banner). While the master switch is off — or the licence is missing/invalid — the module emits nothing by design.

### I see duplicate Product / OpenGraph tags.

Another SEO extension (e.g. Mirasvit) is also emitting structured data. Run only one source — either disable the other extension's snippets, or turn this module's off — otherwise search engines and social scrapers may pick the wrong data.

### The brand is empty in the Product JSON-LD.

The Brand attribute code points at an attribute that's empty or doesn't exist. Set it to your real brand attribute (e.g. manufacturer). Placeholder values like UNKNOWN / NON-BRANDED are intentionally treated as no brand.

### gtin13 doesn't appear.

Either the GTIN-13 attribute code is blank (default), or the product has no barcode. Point it at your barcode/EAN attribute; products without a barcode correctly emit no gtin13 (Google doesn't require one).

### Does it work on Luma?

This module is **Hyvä-native** — it's built for Hyvä storefront themes. Run it on a Hyvä theme.

### Item condition always says New.

That's the default when the Item condition attribute code is blank. Point it at an attribute whose value contains used / refurbished / damaged to map those conditions automatically.

### The category page shows a Breadcrumb but I want the product list too.

Make sure Category page → Enable category rich snippet = Yes; the ItemList is built from the products shown on the current listing page.

### How do I confirm Google will accept it?

Paste the JSON-LD block into validator.schema.org (works on any URL, even password-protected via copy-paste). Once the site is public, re-run Google's Rich Results Test for the live confirmation.