

Best Practices

Advanced Breadcrumbs for Magento 2

Breadcrumbs are a small detail with a disproportionate SEO impact. When configured correctly, they improve crawlability, reinforce site hierarchy in rich results, and reduce bounce rates by giving users a clear sense of where they are. This guide covers recommended configurations based on real-world experience with the [Advanced Breadcrumbs module for Magento 2](#) — including common scenarios, patterns that work well at scale, and mistakes that are easy to make but hard to undo once your catalog grows.

General Guidelines

Do's

- ❑ Keep "Deepest Path" as strategy unless you have a specific reason to change it
- ❑ Set negative weights on non-product categories like "Sale", "New Arrivals", "Clearance" — these are marketing categories, not structural ones
- ❑ Enable JSON-LD structured data — it's the main way search engines pick up breadcrumbs for rich results
- ❑ Use product-level overrides sparingly — only for edge cases where the automatic system fails
- ❑ Test breadcrumbs by viewing page source (not just visually) to confirm they're server-rendered
- ❑ Set custom labels per store view for multi-language stores

Don'ts

- ❑ Don't set weights on every category — only use weights where the default strategy picks the wrong path
- ❑ Don't use "First Assigned" strategy — it's unpredictable and depends on internal Magento sort order
- ❑ Don't disable JSON-LD unless you have another structured data source for breadcrumbs
- ❑ Don't forget to exclude non-structural categories (Sale, Seasonal, Brand-only) with negative weights
- ❑ Don't enable breadcrumbs on every page type just because you can — cart and checkout breadcrumbs add little SEO value

Common Scenarios

Scenario 1: Standard E-commerce Store

Use case: Single-language store with a straightforward category tree (Clothing > Men/Women > Subcategories)

Configuration:

General:

- Enable Module: Yes

Display Settings:

- Show Home Crumb: Yes
- Home Label: Home
- Show Current Page: Yes
- Separator: >

Structured Data:

- Enable JSON-LD: Yes
- Include Home in JSON-LD: Yes

Category Selection:

- Strategy: Deepest Path
- Exclude Non-Anchor: No

Page Types:

- Enabled: Category Pages, Product Pages, CMS Pages

Result: Clean breadcrumbs like Home > Clothing > Men > Jackets > Product Name. JSON-LD generates matching structured data. No weights needed — deepest path handles everything automatically.

Scenario 2: Store with Marketing Categories

Use case: Store has both structural categories (Clothing > Shoes > Sneakers) and marketing categories (Sale, New In, Best Sellers, Gift Ideas)

Configuration:

Same as Scenario 1, plus:

Category weights:

- "Sale" category: weight -1 (excluded)
- "New In" category: weight -1 (excluded)
- "Best Sellers" category: weight -1 (excluded)
- "Gift Ideas" category: weight -1 (excluded)
- All structural categories: weight 0 (default, uses strategy)

Result: Products in both "Sneakers" and "Sale" will always show Home > Clothing > Shoes > Sneakers > Product, never Home > Sale > Product. Marketing categories are excluded from breadcrumb consideration entirely.

Scenario 3: Multi-Language Store

Use case: Store with EN, NL, DE store views

Configuration:

Same base config, plus per store view:

EN store view:

- Home Label: Home
- Cart Label: Shopping Cart
- Account Label: My Account

NL store view:

- Home Label: Home
- Cart Label: Winkelwagen
- Account Label: Mijn Account

DE store view:

- Home Label: Startseite
- Cart Label: Warenkorb
- Account Label: Mein Konto

Result: Breadcrumbs display in the correct language per store view. Category names are already translated via Magento's category store view data.

Scenario 4: Complex Catalog with Priority Categories

Use case: Large store where products belong to many categories and you want specific categories to take priority (e.g., brand categories should always win over generic ones)

Configuration:

Category weights:

- "Nike" (brand category): weight 100
- "Adidas" (brand category): weight 100
- "Running Shoes" (type category): weight 50
- "Men's Shoes" (gender category): weight 25
- "Sale": weight -1

Strategy: Highest Weight (since you're actively managing weights)

Result: A Nike running shoe shows Home > Nike > Running Shoes > Product Name (brand-first hierarchy). Without weights, deepest path might pick a different route.

Scenario 5: SEO-Focused Configuration

Use case: Store where SEO is the top priority, want maximum search engine visibility

Configuration:

- Strategy: Deepest Path (gives most hierarchy context)
- JSON-LD: Enabled
- Include Home in JSON-LD: No (Google recommends omitting home for cleaner rich results — though including it is also fine)
- Show Current Page: Yes (final breadcrumb confirms to users and crawlers what page they're on)
- Page Types: Category, Product, CMS (skip cart/checkout/account — these are noindex pages anyway on most stores)
- Set negative weights on all non-structural categories

Result: Clean, consistent, deep breadcrumb paths across all indexable pages. Structured data matches visual breadcrumbs exactly.

Common Mistakes

Mistake: Setting weights on ALL categories

Why it's wrong: If every category has a weight, you're micromanaging instead of letting the strategy work. It becomes unmaintainable as your catalog grows.

Correct approach: Only set weights where the default strategy picks the wrong path. Use negative weights to exclude non-structural categories. Leave everything else at 0.

Mistake: Using "Highest Weight" strategy without setting weights

Why it's wrong: If no categories have weights (all are 0), this strategy falls back to deepest path silently. It's confusing and means you have a misleading config.

Correct approach: Only use "Highest Weight" if you've actually set meaningful weights on your categories. Otherwise use "Deepest Path".

Mistake: Enabling breadcrumbs on all page types

Why it's wrong: Cart, checkout, and account pages are usually noindex. Breadcrumbs there add visual noise without SEO benefit.

Correct approach: Enable breadcrumbs on Category, Product, and CMS pages. Only add other page types if they're indexed and you want navigation context there.

Mistake: Different structured data sources

Why it's wrong: If you have another module also outputting BreadcrumbList JSON-LD (e.g., a rich snippets module), you'll get duplicate structured data. Google may ignore both or pick the wrong one.

Correct approach: Disable breadcrumb output in other modules. Use Advanced Breadcrumbs as the single source for both visual breadcrumbs and JSON-LD.

Need More Help?

Documentation:

- [All Help Articles](#) - Complete documentation overview

Support:

- [Contact Support](#) - Get help from our team

For a complete overview of features and configuration options, see the Advanced Breadcrumbs extension on magmodules.eu

All articles for Advanced Breadcrumbs

Installation

1	Installatie met Composer (aanbevolen)
2	Installatie via de Adobe Marketplace
3	Installatie via FTP en SSH

Configuration

1	Configuratiegids
---	----------------------------------

Troubleshooting

1	Probleemoplossing
---	-----------------------------------

Usage Guide

1	Snelstartgids
---	-------------------------------

Background

1	Best Practices (current)
2	Breadcrumbs en SEO

