

EmpowerNow Marketing Site Authoring Guide (Astro)

This guide explains how our Astro site is structured and how to add or update content with production-quality SEO, analytics, and NeonFlux design.

1) Project layout

- `src/layouts/`
 - `Base.astro` : global `<head>` , nav, footer, consent + analytics click handler
 - `Product.astro` : product marketing layout (front-matter props below)
 - `Solution.astro` : outcome/solution layout (front-matter props below)
 - `Primer.astro` : standards primers (optional; follows same patterns)
- `src/components/` : reusable UI (LogoWall, PricingCTA, CrossSiteLinks, RelatedContent, JSON-LD components, etc.)
- `src/pages/` : route-based pages (Astro). Key folders:
 - `products/` : one file per product (e.g., `aria-shield.astro`)
 - `solutions/` : one file per solution (`zero-token-spas.astro` , ...)
 - `comparisons/` : programmatic competitor comparisons (`[competitor].astro`)
 - `industries/` : programmatic industry pages (`[industry].astro`)
 - `search/` : Pagefind client search
 - `sitemap*.ts` : dynamic sitemaps per type
- `src/lib/` :
 - `url.ts` : slug/canonical helpers
 - `analytics.ts` : event taxonomy (client-side, consent-gated)
 - `experiments.ts` : simple deterministic A/B helper
 - `keywords.ts` : keyword→related link map
- `public/` : static assets

2) Content types and required props

Use layouts to ensure consistency and SEO.

Product pages (`Product.astro`)

Provide these props in the page file (top of the component):

- `title` (string): HTML/OG title
 - `description` (string): meta description
 - `canonical` (string): canonical path (with trailing slash)
 - `headline` (string): H1
 - Optional: `subhead` (string), `keyBenefits` (string[]), `targetICP` (string[])
 - Optional: `primaryCTA` , `secondaryCTA` ({ `label`, `href` })
 - Optional: `jsonLdProduct` (object) → passed to `<JsonLd />`
 - Optional: `docsLinks` , `wwwLinks` (Link[]) → auto renders `CrossSiteLinks`
 - Optional: `keywords` (string[]) → powers `RelatedContent` via `keywords.ts`
- Slots (named): `how` , `proof` , `related`

SEO/Schema:

- Product pages auto render Breadcrumbs JSON-LD
- Include `jsonLdProduct` with `name/brand/url/description`

Analytics/UX:

- CTAs are annotated with `data-ev` and consent-gated tracking is handled in `Base.astro`
- Experiments: use `getVariant('key', Astro.url.pathname)` and vary CTA labels, etc.

Solution pages (`Solution.astro`)

Required props: `title` , `description` , `canonical` , `headline`

Optional: `outcomes` (string[]), `faq` ({`q,a`}[]), `primaryCTA` , `secondaryCTA` , `docsLinks` , `wwwLinks` , `keywords`

Slots: `how` , `standards` , `related`

SEO/Schema:

- FAQ JSON-LD auto generated when `faq` is provided
- Breadcrumbs JSON-LD is auto rendered

Programmatic pages

- `comparisons/[competitor].astro` : fills a table + JSON-LD (Article) and breadcrumbs
- `industries/[industry].astro` : headline, outcomes, and cross-links

3) Where new content belongs

- New product feature or service → `src/pages/products/<slug>.astro` using `Product.astro`
- New solution/outcome narrative → `src/pages/solutions/<slug>.astro` using `Solution.astro`
- Competitive take/compare → `src/pages/comparisons/[competitor].astro`
- Industry-specific messaging → `src/pages/industries/[industry].astro`
- Standards primer → `Primer.astro` (or keep on `www` `resources/standards` hub if HTML)

Decision rules (updates vs new page):

- Update existing page if the content is the same product/solution and expands clarity, capabilities, or links.
- Create a new page if the audience, intent, or URL taxonomy changes (e.g., a distinct solution outcome, a new service, or a new competitor/industry).
- Keep URLs short, lowercase, hyphenated; end with trailing slash. Update sitemaps happen automatically via dynamic sitemap handlers.

4) NeonFlux styling

- Use layout defaults and utilities in `src/styles/tokens.css` and `src/styles/utilities.css`
- Prefer `.glass-card`, `.btn`, `.btn-primary`, `.grid-3`, `.container`
- For homepage/hero, wrap with `.hero-ambient` to add subtle layered gradients.
- Use `.section-divider` between major sections to create rhythm without heavy borders.
- Avoid inline colors; when needed use `color:#9ba3b7` for secondary text

5) SEO and structured data

- Always set `title`, `description`, and `canonical`
- Product pages: include JSON-LD Product via `<JsonLd data={jsonLdProduct} />`
- FAQ: supply `faq` to `Solution.astro` to auto render FAQPage JSON-LD
- Breadcrumbs JSON-LD is added by layouts
- Sitemaps are dynamic: products, solutions, comparisons, industries

6) Analytics & consent

- Events: add `data-ev`, `data-label`, `data-position` to interactive elements; the global handler in `Base.astro` sends events only after consent

- Consent banner: built-in; users can Allow/Decline; state in `localStorage`
- For server-side tagging, wire your provider to consume `window.dataLayer`

7) Search

- Pagefind runs postbuild: `pagefind --site dist`
- Search UI is at `/search/` ; uses local `/pagefind/pagefind.js`

8) Creating a new page (checklist)

1. Choose layout (`Product.astro` OR `Solution.astro`)
2. Fill required props: `title` , `description` , `canonical` , `headline`
3. Add `docsLinks` and `wwwLinks` for cross-site discoverability
4. Add `keywords` for `RelatedContent`
5. Add `keyBenefits` or `outcomes` and slots (`how` , `proof` , etc.)
6. For product: include `jsonLdProduct`
7. Ensure CTAs include `data-ev` attributes for analytics
8. Verify build: `npm run build` (CI also runs Lighthouse + Pa11y)

9) Editing existing content

- Update the page file in `src/pages/...`
- Keep canonical URL stable unless the taxonomy truly changes; if changing, add a redirect row in `src/pages/_redirects.csv` (`from,to,code`)
- Validate links; use `CrossSiteLinks` to ensure docs/www bi-directional navigation

10) CI and quality gates

- On PRs/commits, GitHub Actions build the site, run Lighthouse (with budgets) and Pa11y a11y checks
- Staging/preview get `X-Robots-Tag: noindex` via middleware

11) For AI contributors (Cursor)

- Determine target layout: product vs solution vs comparison vs industry

- If content matches an existing page's purpose, edit that page; otherwise create a new page with a new slug
- Always set `title` , `description` , `canonical` ; add JSON-LD where applicable
- Add cross-links: set `docsLinks` , `wwwLinks` , and `keywords`
- Keep NeonFlux classes and avoid introducing one-off inline styles
- After changes: add to git, commit with a clear message, and push

12) Commands

- Dev: `npm run dev`
- Build: `npm run build` (then Pagefind runs via `postbuild`)
- Preview: `npm run preview`

Appendix: Front-matter examples live in `src/pages/products/aria-shield.astro` and `src/pages/solutions/zero-token-spas.astro` .

25) Web developer workflow (branches, PRs, local QA)

- Branch naming: `feat/<area>-<slug>` , `fix/<area>-<slug>` , `docs/<area>-<slug>`
- Commit format: `<scope>: <concise change>` (e.g., `site: add Spend Governance solution page`)
- PR checklist:
 - Build passes locally (`npm ci && npm run build`) and preview works (`npm run preview`)
 - Lighthouse local run \geq budgets; Pa11y threshold not exceeded
 - Links resolve; no `.md` links; trailing slashes
 - Sitemaps include new pages (handlers are dynamic)
 - Redirects added if URLs changed (`src/pages/_redirects.csv`)
 - JSON-LD present as applicable (Product, FAQ, Breadcrumbs, Article/HowTo/Video)
- Local commands:
 - Serve dist for audits: `npx http-server dist -p 4321`
 - Lighthouse: `npx lhci autorun --collect.url=http://localhost:4321/`
 - Pa11y: `npx pa11y http://localhost:4321/ --threshold 5`

26) Coding standards (Astro/TS/a11y)

- TypeScript/Astro
 - Prefer explicit props interfaces; meaningful names

- Single responsibility per component; compose small parts
- Avoid inline JS for complex logic—extract to `src/lib/*`
- Formatting
 - Keep lines readable; break long attribute lines
 - Consistent attribute order: `class`, `href/src`, `aria-`, *`data-`*, `style`
- Accessibility
 - One `<h1>` per page; logical heading order
 - `alt` text is mandatory and descriptive
 - Buttons vs links: actions use `<button>`, navigation uses `<a>`
 - Focus states remain visible; don't disable outlines

27) Assets and images

- Store public assets under `public/` when shared; page-local assets can live adjacent to pages
- Formats: SVG for logos/icons; PNG or optimized JPEG/WebP for photos
- Size caps: keep inline images < 200 KB; prefer `ResponsiveImg.astro`
- Favicon/OG updates: update `public/favicon.svg` and use `/og/[text].svg`

28) Environments and redirects

- Preview/stage: use `npm run preview`; staging gets `X-Robots-Tag: noindex` via middleware
- Redirect process: create or update rows in `src/pages/_redirects.csv` and test locally

29) Error playbook

- Pagefind search returns no results
 - Ensure `npm run build` ran and `postbuild` generated the index; verify `/pagefind/pagefind.js` in `dist`
- Sitemap empty for a section
 - Confirm files exist and globs in `src/pages/sitemaps/*.ts` match `(**/*.astro,md,mdx)`
- 404 on internal links
 - Add missing trailing slash; ensure path is under `src/pages/`; no `.md` in URLs
- Build fails
 - Check TypeScript imports/paths, missing props in layout usage, or stray JSX syntax

30) Marketing assistant workflow

- Content brief template
 - Audience: ...
 - Outcome/value: ...
 - Proof/evidence (data, receipts, quotes): ...
 - Primary CTA / Secondary CTA: ...
 - Target keywords (2–4) + internal links to include: ...
- Voice and style guardrails
 - Headline: outcome-led, ≤ 9 words; Subhead: how it works in one line
 - Use short paragraphs and 2–4 bullet lists; avoid buzzwords; prefer concrete benefits
- CTA rules
 - Primary CTA is action verb + outcome (e.g., “Book demo”); Secondary is “Read docs”/“See deep dives”
 - Add `data-ev` , `data-label` , `data-position` via layout props
- SEO workflow
 - Put primary keyword in `title` , H1, first paragraph; link to 2+ internal relevant pages
 - Choose schema: Product for product pages, FAQ on Q&A, Article/TechArticle for educational pieces
- Images and OG
 - Provide alt text and, if social share expected, specify OG text for `/og/[text].svg`
- Approval flow
 - SME review (factual), Marketing review (voice/SEO), Dev review (structure/links), then merge

31) Page scaffolds (copy/paste and fill)

Minimal Product page

```

---
import Base from '../..../layouts/Base.astro';
import Product from '../..../layouts/Product.astro';
import JsonLd from '../..../components/JsonLd.astro';
const title = 'Product – One-liner';
const description = 'Short meta description.';
const canonical = '/products/product-slug/';
const jsonLdProduct = {"@context":"https://schema.org","@type":"Product","name":"Product","brand":
---
<Base title={title} description={description} canonical={canonical}>
  <JsonLd data={jsonLdProduct} />
  <Product title={title} description={description} canonical={canonical} headline="Outco
    <fragment slot="how">
      <section class="container" style="padding:8px 0 24px">
        <h2 class="section-title">How it works</h2>
        <p style="color:#9ba3b7">Short explainer...</p>
      </section>
    </fragment>
  </Product>
</Base>

```

Minimal Solution page


```

---
import Base from '../layouts/Base.astro';
import Solution from '../layouts/Solution.astro';
const title = 'Solution – Outcome';
const description = 'Short meta description.';
const canonical = '/solutions/solution-slug/';
---
<Base title={title} description={description} canonical={canonical}>
  <Solution title={title} description={description} canonical={canonical} headline="Outc
    <fragment slot="how">
      <section class="container" style="padding:8px 0 24px">
        <h2 class="section-title">How it works</h2>
        <ol class="numbered">
          <li>Step 1</li>
          <li>Step 2</li>
          <li>Step 3</li>
        </ol>
      </section>
    </fragment>
  </Solution>
</Base>

```

32) Issue/PR templates (suggested)

New page request (marketing)

- Page type: Product | Solution | Comparison | Industry | Primer
- Proposed URL slug:
- Audience & outcome:
- Key benefits/outcomes (3 bullets):
- Proof/evidence links:
- Primary/Secondary CTAs:
- Keywords + internal links:
- Images/OG needs:

Edit existing page (dev)

- File(s) to edit:
- Change summary:
- URL changes (if any) + redirects added:
- JSON-LD updates:
- Cross-links added (docs/www):
- QA: Lighthouse/Pa11y status:

33) Component catalog (when to use)

- LogoWall: social proof row for known brands
- TestimonialCard: short quotes with author/role
- TrustBadges: certifications/compliance chips
- ComparisonTable: side-by-side capability compare
- CrossSiteLinks: render docs/www link blocks
- RelatedContent: auto links sourced from keywords
- PricingCTA: wide call-to-action bar

34) Router mapping (what goes where)

- Product → /products/<slug>/ → Product.astro → required: title, description, canonical, headline
- Solution → /solutions/<slug>/ → Solution.astro → required: title, description, canonical, headline
- Comparison → /comparisons/<competitor>/ → programmatic
- Industry → /industries/<industry>/ → programmatic
- Trust pages → /trust/... → simple sections with Base.astro

35) Quality bar (pre-publish)

- H1 present and unique; title/meta/canonical set
- JSON-LD valid for page type(s)
- Internal links and CTAs work; no .md or missing trailing slashes
- Cross-links to docs/www added; keywords render RelatedContent
- a11y: alt text, headings order, focus states
- Performance: images optimized; budgets pass
- Consent banner visible for new sessions