

Astro Blog Project Structure

Content collections with MDX, type-safe frontmatter, and RSS feed.

#astro #blog #mdx #content-collections #rss

PNG

PDF

Copy

</> Prompt

Project Directory

my-blog/

- >

src/

>

components/

Header.astro

Footer.astro

PostCard.astro

TableOfContents.astro

Tags.astro

>

content/

Content collections

>

blog/

Blog posts

first-post.mdx

second-post.mdx

>

authors/

Author profiles

jane-doe.md

config.ts

Collection schema

>

layouts/

Base.astro

Post.astro

Blog post layout

>

pages/

index.astro

Post listing

>

blog/

[...slug].astro

Dynamic post page

>

tags/

index.astro

All tags

[tag].astro

Posts by tag

rss.xml.ts

RSS feed endpoint

>

styles/

global.css

>

public/

favicon.svg

images/

Blog images

astro.config.mjs

package.json

tsconfig.json

.gitignore

Why This Structure?

Content collections in `src/content/` provide type-safe frontmatter via Zod schemas. MDX allows embedding components in Markdown. The `[...slug].astro` pattern generates pages from collection entries. RSS is built-in with `@astrojs/rss`.

Key Directories

- `src/content/` - Type-safe content collections
- `src/content/config.ts` - Zod schemas for frontmatter
- `src/pages/blog/[...slug].astro` - Dynamic routes from collection
- `rss.xml.ts` - Auto-generated RSS feed

Content API

- `defineCollection()` - Define schema in `config.ts`
- `getCollection()` - Query all entries type-safe
- `getEntry()` - Get single entry by slug
- `render()` - Compile MDX to component

Getting Started

```
npm create astro@latest -- --template blog
npm install @astrojs/mdx @astrojs/rss
npm run dev
```

When To Use This

- Personal or company blogs
- Documentation sites with structured content
- Sites needing RSS feeds
- Content-heavy sites with multiple authors

When To Upgrade

- Need real-time or user-generated content
- Require authentication or user sessions
- Want interactive dashboards