Content and Data Layer

The site has two kinds of data. Both are read **at build time** only.

Kind	Location	Edited by	Loaded via
CMS content	`content/*.json`	Staff via Decap, or developers	Static JSON imports in `src/lib/*` loaders
Structured code content	`src/lib/*.ts` exports	Developers	Direct imports

CMS content files

File	Shape	Loader	Consumers
`content/alert.json`	`{ enabled: boolean, message: string }`	`getSiteAlert()` in `content.ts`, which returns `enabled` only if the trimmed message is non-empty	`AlertBanner` (root layout)
`content/tour-booking.json`	`{ bookingUrl: string, guidedBookingUrl: string }`	`getTourBooking()`. Both are validated with `isFareharborBookingUrl()` (https, host `fareharbor.com`, path starts `/embeds/book/`). An invalid default becomes `""`; an invalid or empty guided URL falls back to the default.	Header, Visit page, root layout (loads the script only if a URL exists)
`content/events.json`	`{ events: StoredFortEvent[] }`	`getPublicEvents()`, `getUpcomingEvents(limit?)`, `getFeaturedEvent()`	Home, Events
`content/event-categories.json`	`{ categories: { name }[] }`	`EVENT_CATEGORIES` (trimmed, deduped, order preserved)	Events filters
`content/staff.json`	`{ members: StaffMember[], board: BoardMember[] }`	`getStaff()`, `getBoard()`	About
`content/sponsors.json`	`{ year, heading, intro, sponsors: CorporateSponsor[] }`	`getSponsorDirectory()` (drops `isPublished === false`)	Support Us

> Loaders use type assertions, not runtime schema validation. Decap enforces field rules inthe editor, but hand-edited JSON isn't checked. Run `npm run build` after editing `content/` by hand.

The events pipeline

```
interface FortEvent {
  id: string;            // optional in JSON; generated if missing
  title: string;
  description: string;
  startDate: string;     // ISO 8601 with offset, e.g. "2026-10-03T09:30:00-04:00"
  endDate: string;       // "" allowed → treated as startDate
  location: string;
  imageUrl: string;      // "" → FortImage fallback
  isPublished?: boolean; // omitted = visible
  isFeatured: boolean;
  category: string;
  externalUrl?: string;  // → "Get tickets" (new tab); takes precedence
  bookingUrl?: string;   // site path → "Book now"
}
```

`getPublicEvents()` filters out `isPublished === false`, fills in missing IDs (`slug(startDate + "-" + title)`, since Decap's `id` field is hidden and never set for new entries), and sorts by `startDate`.

`getUpcomingEvents()` keeps events whose `endDate || startDate` is at or after **`Date.now()` at build time**. `getFeaturedEvent()` returns the first upcoming event with `isFeatured`.

Time zone handling. Display helpers (`monthKey`, `formatEventWhen`, `eventDateKey`) all format in `America/New_York`, so labels don't depend on the build machine or the visitor's time zone. `formatEventWhen` shows a date range for multi-day events, and omits the time when the start is exactly midnight Eastern ("all-day").

Calendar. `event-calendar.ts` builds a Sunday-first month grid. Multi-day events appear on every day in their range.

Structured code content (`src/lib`)

Module	Exports	Drives

`site-config.ts`	`siteConfig` (name, URL, org address/phone/email/EIN, social), `primaryNav`, `cta`, `footerLinks`	Header, footer, metadata, sitemap, DAF panel, form error messages
`tours.ts`	`tours: Tour[]` (guided, self-guided, private-group)	Visit tour cards (copy only; FareHarbor holds the real products)
`faq.ts`	`visitFaqGroups: FaqGroup[]` (3 groups, about 30 Q&As)	Visit FAQ accordion
`venues.ts`	`venues: Venue[]`	Host venue cards
`host-content.ts`	`eventTypes`, `inquiryEventTypes`, `guestCountOptions`, `budgetOptions`, `whatsIncluded`, `whyFortAdams`, `hostTestimonials` (**unused placeholder**)	Host page and inquiry form
`support-content.ts`	`membershipTiers` (with HighLevel payment-link URLs), `impactStats`, `isJoinable()`, plus legacy `donationPresets` / `donationImpact()` (unused since the hosted-form switch)	Support Us
`history.ts`	`historyChapters`, `mapHotspots`, `restorationProjects`	History page. Supports any number of chapters and hotspots; `mapId` links a chapter to a hotspot.
`fort-terminology.ts`	`architectureFeatures`, `fortTerms` (about 100)	Architecture & Terminology
`explore-fort.ts`	`exploreVideos` (exactly 6), `getExploreMediaUrl()`	Home Explore the Fort. **Throws at build** if not exactly six complete, unique entries with 0–100 coordinates.

Placeholders

Content awaiting the client is marked `[PLACEHOLDER: …]`. Find it all with:

```
grep -rn "PLACEHOLDER" src content public/admin functions
```

Revision #1
Created 2026-09-28 14:25:57 UTC by David Prata
Updated 2026-09-28 14:25:57 UTC by David Prata