

Content Management Guide - Atelier Agency Theme

Overview

This guide helps you manage and update content in your Atelier Agency theme. Whether you're working with static data files or integrating with a headless CMS, this document provides clear instructions for content management.

Content Structure

Data Organization

```
/data/  
├── projects.ts      # Portfolio projects  
├── team.ts         # Team members  
├── services.ts     # Service offerings  
├── testimonials.ts # Client testimonials  
├── clients.ts      # Client logos  
└── site-config.ts  # Global site settings
```

Portfolio Management

Adding Projects

Edit `/data/projects.ts` to add or modify portfolio projects:

```

export interface Project {
  id: number
  title: string
  category: string
  description: string
  image: string
  gallery?: string[]
  technologies: string[]
  client: string
  year: string
  url?: string
  featured: boolean
  status: 'published' | 'draft' | 'archived'
}

export const projects: Project[] = [
  {
    id: 1,
    title: "Architectural Visualization Platform",
    category: "Web Development",
    description: "A cutting-edge platform for architects to showcase 3D visualizations with interactive elements and immersive experiences.",
    image: "/images/projects/arch-viz-hero.jpg",
    gallery: [
      "/images/projects/arch-viz-detail-1.jpg",
      "/images/projects/arch-viz-detail-2.jpg",
      "/images/projects/arch-viz-detail-3.jpg"
    ],
    technologies: ["Next.js", "Three.js", "TypeScript", "GSAP"],
    client: "Nordic Architecture Studio",
    year: "2024",
    url: "https://nordic-arch.com",
    featured: true,
    status: 'published'
  }
]

```

Project Categories

Standardize categories for consistent filtering:

```

export const projectCategories = [
  "Web Development",
  "Brand Identity",
  "UI/UX Design",
  "Mobile App",
  "E-commerce",
  "Digital Strategy",
  "3D Visualization",
  "Print Design"
]





```

Image Guidelines

Project Images:

- **Hero Image:** 1920×1080px (16:9)
- **Gallery Images:** 1600×1200px (4:3)
- **Thumbnail:** 800×600px (4:3)

- **Format:** WebP preferred, JPG fallback
- **Optimization:** Use Next.js Image component

```
// Image naming convention
/public/images/projects/
 project-slug-hero.jpg
 project-slug-gallery-1.jpg
 project-slug-gallery-2.jpg
 project-slug-thumbnail.jpg
```

Team Management

Adding Team Members

Update `/data/team.ts` :

```
export interface TeamMember {
  id: number
  name: string
  role: string
  image: string
  bio: string
  skills: string[]
  social: {
    linkedin?: string
    twitter?: string
    instagram?: string
    dribbble?: string
    behance?: string
  }
  featured: boolean
}

export const team: TeamMember[] = [
  {
    id: 1,
    name: "Alexandra Chen",
    role: "Creative Director",
    image: "/images/team/alexandra-chen.jpg",
    bio: "With over 10 years of experience in digital design, Alexandra leads our creative vision with architectural precision and minimalist aesthetics.",
    skills: ["Creative Direction", "Brand Strategy", "Design Systems", "Team Leadership"],
    social: {
      linkedin: "https://linkedin.com/in/alexandra-chen",
      dribbble: "https://dribbble.com/alexandra-chen"
    },
    featured: true
  }
]
```

Team Photo Guidelines

- **Size:** 800×800px (1:1 square)
- **Style:** Professional, consistent lighting
- **Background:** Neutral or removed
- **Format:** WebP preferred

- **Naming:** `firstname-lastname.jpg`

Services Management

Service Configuration

Edit `/data/services.ts` :

```
export interface Service {
  id: number
  title: string
  description: string
  features: string[]
  icon: string
  category: 'design' | 'development' | 'strategy' | 'branding'
  pricing?: {
    starting: number
    currency: string
  }
  featured: boolean
}

export const services: Service[] = [
  {
    id: 1,
    title: "Web Development",
    description: "Custom websites built with modern technologies and architectural precision.",
    features: [
      "Responsive Design",
      "Performance Optimization",
      "SEO Implementation",
      "CMS Integration",
      "E-commerce Solutions"
    ],
    icon: "code",
    category: "development",
    pricing: {
      starting: 5000,
      currency: "USD"
    },
    featured: true
  }
]
```

Client & Testimonial Management

Client Testimonials

Update `/data/testimonials.ts` :

```

export interface Testimonial {
  id: number
  content: string
  author: {
    name: string
    role: string
    company: string
    image: string
  }
  project?: string
  rating: number
  featured: boolean
}

export const testimonials: Testimonial[] = [
  {
    id: 1,
    content:
      "The team at Atelier delivered beyond our expectations. Their attention to detail and architectural approach to web design is unmatched.",
    author: {
      name: "Marcus Rodriguez",
      role: "CEO",
      company: "Zenith Architects",
      image: "/images/testimonials/marcus-rodriguez.jpg"
    },
    project: "Architectural Visualization Platform",
    rating: 5,
    featured: true
  }
]

```

Client Logos

Manage client logos in `/data/clients.ts`:

```

export interface Client {
  id: number
  name: string
  logo: string
  website?: string
  industry: string
}

export const clients: Client[] = [
  {
    id: 1,
    name: "Nordic Architecture Studio",
    logo: "/images/clients/nordic-arch-logo.svg",
    website: "https://nordic-arch.com",
    industry: "Architecture"
  }
]

```

Site Configuration

Global Settings

Configure site-wide settings in `/data/site-config.ts`:

```

export const siteConfig = {
  name: "Atelier Agency",
  description: "Digital Stonemasonry for Modern Brands",
  url: "https://atelier-agency.com",

  // Contact Information
  contact: {
    email: "hello@atelier-agency.com",
    phone: "+1 (555) 123-4567",
    address: {
      street: "123 Design Street",
      city: "San Francisco",
      state: "CA",
      zip: "94102",
      country: "USA"
    }
  },

  // Social Media
  social: {
    twitter: "https://twitter.com/atelier_agency",
    linkedin: "https://linkedin.com/company/atelier-agency",
    instagram: "https://instagram.com/atelier_agency",
    dribbble: "https://dribbble.com/atelier_agency"
  },

  // Business Hours
  hours: {
    monday: "9:00 AM - 6:00 PM",
    tuesday: "9:00 AM - 6:00 PM",
    wednesday: "9:00 AM - 6:00 PM",
    thursday: "9:00 AM - 6:00 PM",
    friday: "9:00 AM - 5:00 PM",
    weekend: "Closed"
  },

  // SEO Settings
  seo: {
    title: "Atelier Agency - Digital Stonemasonry",
    titleTemplate: "%s | Atelier Agency",
    description: "Premium web design and development with architectural precision. We craft digital experiences that stand the test of time.",
    keywords: ["web design", "digital agency", "branding", "web development"],
    ogImage: "/images/og-image.jpg"
  }
}

```

Content Management Systems

Headless CMS Integration

The theme is designed to work with popular headless CMS solutions:

Sanity CMS

```
npm install @sanity/client next-sanity
```

Create `/lib/sanity.ts`:

```
import { createClient } from '@sanity/client'

export const sanity = createClient({
  projectId: 'your-project-id',
  dataset: 'production',
  useCdn: true,
  apiVersion: '2023-05-03'
})

// Fetch projects
export async function getProjects() {
  return await sanity.fetch(`
    *[_type == "project"] | order(publishedAt desc) {
      _id,
      title,
      slug,
      description,
      "image": image.asset->url,
      category,
      technologies,
      client,
      year
    }
  `)
}
```

Strapi CMS

```
// lib/strapi.ts
export async function fetchAPI(endpoint: string) {
  const response = await fetch(`${process.env.STRAPI_URL}/api${endpoint}`)
  return response.json()
}

export async function getProjects() {
  const data = await fetchAPI('/projects?populate=*')
  return data.data
}
```

Contentful

```
// lib/contentful.ts
import { createClient } from 'contentful'

const client = createClient({
  space: process.env.CONTENTFUL_SPACE_ID!,
  accessToken: process.env.CONTENTFUL_ACCESS_TOKEN!
})

export async function getProjects() {
  const entries = await client.getEntries({
    content_type: 'project'
  })
  return entries.items
}
```

Content Migration

From Static Files to CMS

1. **Export existing data:**

```
node scripts/export-content.js
```

1. **Create CMS schemas** matching your data structure
2. **Import content** using CMS-specific tools
3. **Update data fetching** in your components

CMS Schema Examples

Project Schema (Sanity)

```
export default {
  name: 'project',
  title: 'Project',
  type: 'document',
  fields: [
    {
      name: 'title',
      title: 'Title',
      type: 'string'
    },
    {
      name: 'slug',
      title: 'Slug',
      type: 'slug',
      options: {
        source: 'title',
        maxLength: 96
      }
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text'
    },
    {
      name: 'image',
      title: 'Featured Image',
      type: 'image'
    },
    {
      name: 'gallery',
      title: 'Gallery',
      type: 'array',
      of: [{ type: 'image' }]
    },
    {
      name: 'category',
      title: 'Category',
      type: 'string',
      options: {
        list: [
          { title: 'Web Development', value: 'web-development' },
          { title: 'Brand Identity', value: 'brand-identity' },
          { title: 'UI/UX Design', value: 'ui-ux-design' }
        ]
      }
    },
    {
      name: 'technologies',
      title: 'Technologies',
      type: 'array',
      of: [{ type: 'string' }]
    }
  ]
}
```

SEO Content Management

Meta Tags

Update page-specific meta tags:

```
// app/work/page.tsx
export const metadata = {
  title: "Our Work - Portfolio",
  description: "Explore our portfolio of premium web design and development projects.",
  openGraph: {
    title: "Our Work - Atelier Agency Portfolio",
    description: "Premium digital experiences crafted with architectural precision.",
    images: ["/images/work-og.jpg"]
  }
}
```

JSON-LD Structured Data

Add structured data for better SEO:

```
// lib/structured-data.ts
export const organizationSchema = {
  "@context": "https://schema.org",
  "@type": "Organization",
  "name": siteConfig.name,
  "url": siteConfig.url,
  "logo": `${siteConfig.url}/logo.svg`,
  "description": siteConfig.description,
  "address": {
    "@type": "PostalAddress",
    "streetAddress": siteConfig.contact.address.street,
    "addressLocality": siteConfig.contact.address.city,
    "addressRegion": siteConfig.contact.address.state,
    "postalCode": siteConfig.contact.address.zip,
    "addressCountry": siteConfig.contact.address.country
  },
  "contactPoint": {
    "@type": "ContactPoint",
    "telephone": siteConfig.contact.phone,
    "contactType": "customer service"
  }
}
```

Image Optimization

Image Processing Pipeline

```
// scripts/optimize-images.js
import sharp from 'sharp'
import fs from 'fs'

async function optimizeImages() {
  const inputDir = './public/images/raw'
  const outputDir = './public/images'

  const files = fs.readdirSync(inputDir)

  for (const file of files) {
    // Generate WebP
    await sharp(`${inputDir}/${file}`)
      .webp({ quality: 85 })
      .toFile(`${outputDir}/${file.replace(/\.[^/.]+$/, '.webp')}`)

    // Generate optimized JPG
    await sharp(`${inputDir}/${file}`)
      .jpeg({ quality: 85, progressive: true })
      .toFile(`${outputDir}/${file}`)

    // Generate thumbnail
    await sharp(`${inputDir}/${file}`)
      .resize(400, 300, { fit: 'cover' })
      .webp({ quality: 80 })
      .toFile(`${outputDir}/thumbnails/${file.replace(/\.[^/.]+$/, '-thumb.webp')}`)
  }
}
```

Content Validation

TypeScript Validation

Use TypeScript interfaces to ensure content consistency:

```
// lib/content-validation.ts
import { z } from 'zod'

export const ProjectSchema = z.object({
  id: z.number(),
  title: z.string().min(1),
  category: z.string().min(1),
  description: z.string().min(10),
  image: z.string().url(),
  technologies: z.array(z.string()),
  client: z.string().min(1),
  year: z.string().length(4),
  featured: z.boolean(),
  status: z.enum(['published', 'draft', 'archived'])
})

export function validateProject(project: unknown) {
  return ProjectSchema.parse(project)
}
```

Content Deployment

Automated Content Updates

Set up GitHub Actions for content deployment:

```
# .github/workflows/content-update.yml
name: Content Update
on:
  push:
    paths:
      - 'data/**'
      - 'public/images/**'

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Deploy to Vercel
        uses: amondnet/vercel-action@v20
        with:
          vercel-token: ${ secrets.VERCEL_TOKEN }
          vercel-args: '--prod'
```

This guide ensures your content management is as precise and efficient as the theme's architecture. For technical implementation details, refer to the Technical Specification document.