

# The Essential Headless CMS Migration Checklist

This checklist serves as a step-by-step guide for teams planning and executing a transition from a traditional (monolithic) CMS to a headless architecture. Use it to track progress, assign tasks, and ensure all critical aspects are covered.

## Phase 1: Strategy & Planning (Pre-Migration)

This phase lays the foundation for the entire project. Mistakes made here can be costly later on.

- **[ ] Define Project Goals:**
    - [ ] Clearly articulate the "why" behind the migration (e.g., improved performance, omnichannel delivery, better developer experience).
    - [ ] Define measurable KPIs to evaluate success (e.g., page load speeds, content time-to-market).
  - **[ ] Assemble the Project Team:**
    - [ ] Appoint a Project Manager.
    - [ ] Identify team members: Frontend Developers, Backend Developers, DevOps Specialist, SEO Specialist, Content Strategist, UX/UI Designer.
  - **[ ] Audit the Existing System:**
    - [ ] Analyze the current architecture, performance, and bottlenecks.
    - [ ] Conduct a full content audit (what to migrate, what to archive, what to delete).
  - **[ ] Select the Technology Stack:**
    - [ ] Choose a suitable Headless CMS (e.g., Strapi, Contentful, Sanity).
    - [ ] Choose a Frontend Framework (e.g., Next.js, Gatsby, Nuxt.js).
    - [ ] Choose a Hosting Provider and CDN (e.g., Vercel, Netlify, AWS).
  - **[ ] Design the Content Model:**
    - [ ] Identify all content types (e.g., articles, authors, products, pages).
    - [ ] Define the fields and data types for each model.
    - [ ] Design the relationships between models (e.g., an article has an author).
  - **[ ] Plan URL Structure & Redirects:**
    - [ ] Create a map of all existing URLs.
    - [ ] Design the new URL structure (if needed).
    - [ ] Create a comprehensive 301 redirect map from old URLs to new ones.
  - **[ ] Establish Budget and Timeline:**
    - [ ] Estimate costs for development, service subscriptions, and maintenance.
    - [ ] Create a realistic project timeline with key milestones.
-

## Phase 2: Technical Setup & Development

This is the stage where the new site's technical foundation is built.

- **[ ] Set Up the Headless CMS:**
  - [ ] Deploy the CMS instance.
  - [ ] Implement the designed content model in the CMS interface.
  - [ ] Configure user roles and permissions (e.g., editors, administrators).
- **[ ] Set Up the Frontend Project:**
  - [ ] Create the code repository and set up the development environment.
  - [ ] Connect the frontend application to the Content API.
  - [ ] Configure environment variables (API keys, endpoints).
- **[ ] Develop Components and Templates:**
  - [ ] Build UI components (e.g., buttons, forms, navigation).
  - [ ] Develop page templates (e.g., homepage, article page, category page).
  - [ ] Implement the logic for fetching and displaying data from the API.
- **[ ] Implement Functionality:**
  - [ ] Set up routing.
  - [ ] Implement site search.
  - [ ] Configure form handling (e.g., contact forms, newsletters).
  - [ ] Set up third-party integrations (e.g., analytics, CRM).
- **[ ] Configure a Preview Mode:**
  - [ ] Implement a system that allows content editors to preview their changes before publishing.

---

## Phase 3: Content Migration

Transferring the actual data from the old platform to the new one.

- **[ ] Choose a Migration Method:**
    - [ ] Manual transfer (for small sites).
    - [ ] Automated (via custom scripts).
    - [ ] Hybrid approach.
  - **[ ] Prepare and Run Scripts (if applicable):**
    - [ ] Write and test migration scripts on a small subset of data.
    - [ ] Execute the full migration of all content.
  - **[ ] Perform Migration QA:**
    - [ ] Verify the correct transfer of text, images, and metadata.
    - [ ] Check for broken internal links and missing images.
    - [ ] Ensure that formatting has been preserved.
-

## Phase 4: Launch (Go-Live)

The final steps before the new site is available to users.

- **[ ] Conduct Final Testing:**
  - [ ] Perform cross-browser and cross-device testing.
  - [ ] Test performance and Core Web Vitals.
  - [ ] Conduct a final SEO audit (check meta tags, sitemap.xml, robots.txt).
  - [ ] Test all forms and interactive elements.
- **[ ] Implement Redirects:**
  - [ ] Deploy and activate the 301 redirect map.
- **[ ] Freeze Content on the Old Site:**
  - [ ] Prohibit any further content changes on the old platform.
- **[ ] Switch DNS:**
  - [ ] Update the domain's DNS records to point to the new host.
  - [ ] Wait for the DNS changes to fully propagate.
- **[ ] Perform Post-Launch Checks:**
  - [ ] Run a site crawler (e.g., Screaming Frog) to find 404 errors and redirect issues.
  - [ ] Verify that Google Analytics and other tracking scripts are firing correctly.

---

## Phase 5: Post-Launch Optimization

Work on the site doesn't end at launch.

- **[ ] Monitoring & Analysis:**
  - [ ] Track site performance and Core Web Vitals.
  - [ ] Monitor indexing status and crawl errors in Google Search Console.
  - [ ] Analyze user behavior and traffic patterns.
- **[ ] Gather Feedback:**
  - [ ] Collect feedback from the content team on their experience with the new CMS.
  - [ ] Analyze user feedback (if applicable).
- **[ ] Plan Iterations:**
  - [ ] Create a plan to address any identified issues.
  - [ ] Plan new feature development based on analytics and feedback.