

The Headless CMS Buyer's Guide

Finding the best Headless
CMS for your business.

Table of Contents

Key takeaways	p. 3
Introduction	p. 4
What is a headless CMS?	p. 4
Why would someone need a headless CMS?	p. 8
Pros of going headless	p. 8
Cons of going headless	p. 9
Use cases of headless	p. 9
Considerations before looking for a CMS	p. 10
Considerations when picking a CMS	p. 12
Questions to ask when considering a headless CMS	p. 16
Headless CMS evaluation checklist	p. 17
Headless CMS usage in real life	p. 18
About ButterCMS	p. 19

Key Takeaways

- A headless CMS is an evolved CMS that decouples the backend code from the frontend layer. It's a backend-only CMS that can serve content to any frontend using APIs. Unlike traditional CMSs, headless lets you go beyond websites, and serve content to any-and-all digital products.
- A headless CMS lets you reuse the same code for multiple touchpoints, enabling you to deliver omnichannel customer experiences. It increases code maintainability, enhances the content management experience, and decreases your time to market. It's an ideal fit for several use cases, including e-commerce websites, digital devices, native mobile apps, and SSGs.
- Before you begin searching for a headless CMS you need to know your team, understand their pain points, and document their requirements. You also need to carefully validate your need for a headless CMS, note down your goals and objectives, Define who your primary stakeholders are, and separate your must-have features from the nice-to-haves.
- There are several things to consider while picking a CMS, including API architecture, SEO support, pricing, content editor, user experience, developer friendliness, security, customizability, hosting, and technical support.
- Headless CMSs have enabled several companies to solve real-world problems. E.g. Freddie's Flowers used ButterCMS to scale their business around the globe. PickFu, an online consumer research company, used ButterCMS to decouple their marketing and dev teams.

Introduction

A content management system (CMS) simplifies and fast-tracks the process of building a website. You don't need to write all the code from scratch, and you don't need to rely on your developers to tweak its design or content.

For decades, traditional CMSs remained the staple of web development, but today, businesses are exploring the enhanced flexibility and usability offered by headless CMSs. Among other things, a headless CMS fixes one inherent flaw in traditional CMSs – it lets you go beyond websites, and deliver an omnichannel customer experience.

If you want to build a single platform to serve content for your websites, mobile applications, conversational UIs, and several other digital touchpoints, then a headless CMS is your best bet. It will organize all your content in one place, and empower you to render it on any frontend you want.

Choosing the right headless CMS for your business is critical. It can also be overwhelming, with so many vendors and factors to consider. But fret not, in this guide, we will break down everything you need to evaluate to choose a fitting headless CMS.

We'll start by taking a deep dive into headless CMS architecture, then talk about things to consider before and while picking a headless CMS, look at a few real-life use-cases, and finish off with a list of leading headless CMSs.

What is a headless CMS?

A headless CMS is a content management system that provides a backend interface which can connect with any frontend. Content stored in your headless CMS can be served via APIs and rendered on virtually any frontend, e.g. website, mobile application, smart watches and assistants etc.

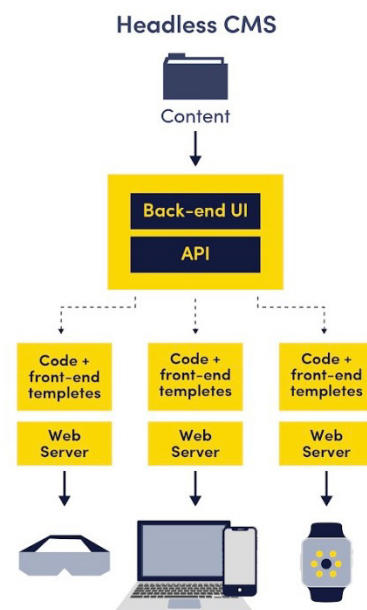
As the content is decoupled from the design, you don't have to tweak your backend code to connect to a new digital touchpoint. This is a gamechanger for businesses that want to implement omnichannel strategies and/or build future-proof, digital products.

Headless CMS architecture

A headless CMS is backend-only. It literally doesn't come with a head, i.e. the presentation layer. But that's a good thing, because you can connect it with any head (frontend), or even many heads! These connections are made using APIs, which is code that lets applications/systems talk to each other and exchange information (in our case, the frontend and backend systems).

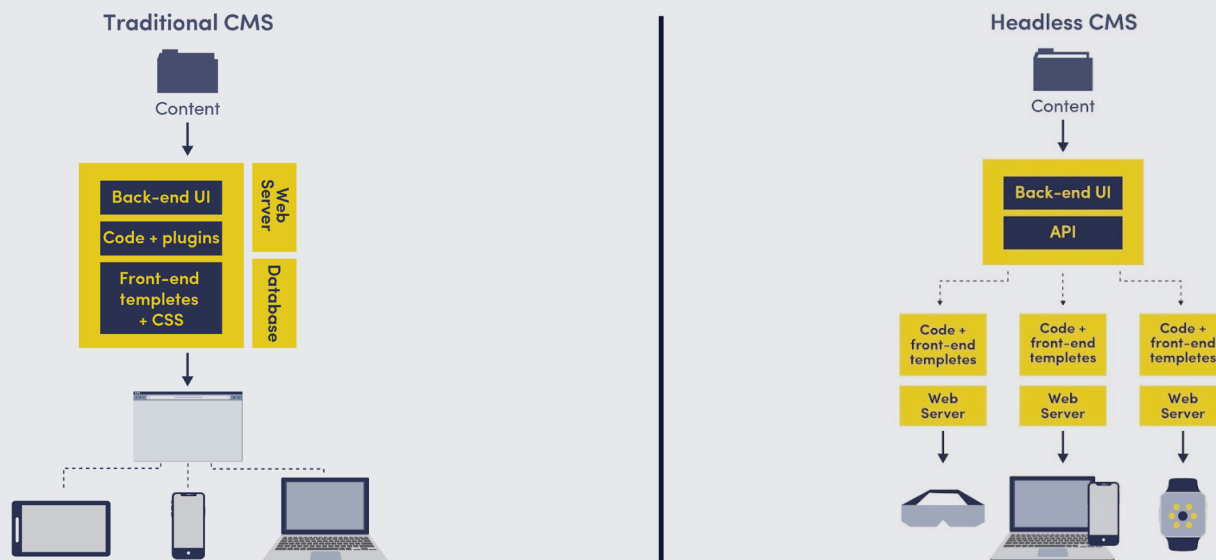
A typical headless CMS architecture consists of:

- A database to store all the content and any digital assets.
- An interface to create/edit/publish content.
- A set of APIs that allows frontend developers to fetch content for different applications.



The evolution of CMSs

Traditional vs. Headless CMS



Over the years, CMSs have evolved as per the changing needs of the internet and its consumers. Traditional CMSs were powerful tools to build and customize websites with lots of text and multimedia. They put the entire website's content in one place, including the source code for the frontend and the backend.

This worked well for a long time, as websites remained the primary customer touchpoint. But as new digital channels and devices entered the scene, customer expectations evolved. Businesses soon realized that to gain a competitive advantage, they must start delivering content via multiple channels.

This is the point where traditional CMSs started falling behind the curve in terms of flexibility and scalability. You couldn't reuse the same backend to serve content to other digital devices/interfaces, because the backend and frontend code was intermingled.

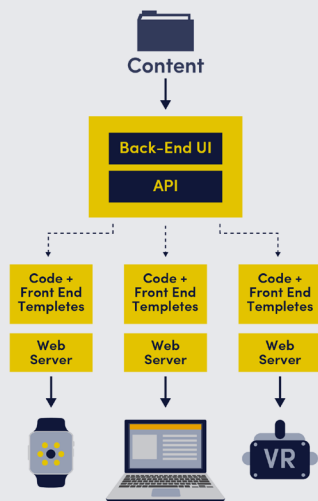
Enter headless CMS: an evolved form of CMS that made it possible to build a single backend for all your frontends. Today, companies can use headless CMS APIs to connect their content repositories with as many digital touchpoints as they like.

Headless CMS vs traditional CMS

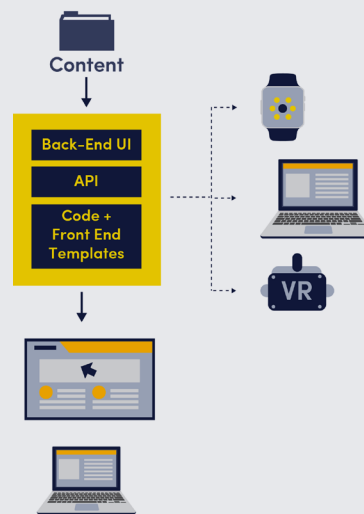
- A traditional CMS uses a monolithic architecture, whereas a headless CMS lets you use a microservices-based approach.
- A traditional CMS can be integrated with limited devices and applications, whereas a headless CMS, through APIs, allows maximum interoperability.
- A traditional CMS organizes content for webpage rendering, whereas a headless CMS provides building blocks for many applications.
- A headless CMS lets you build a proof-of-concept much quicker as compared to a traditional CMS.
- A headless CMS is a future-proof investment for agile teams; a traditional CMS is not.

Headless CMS vs. decoupled (aka hybrid) CMS

Headless CMS

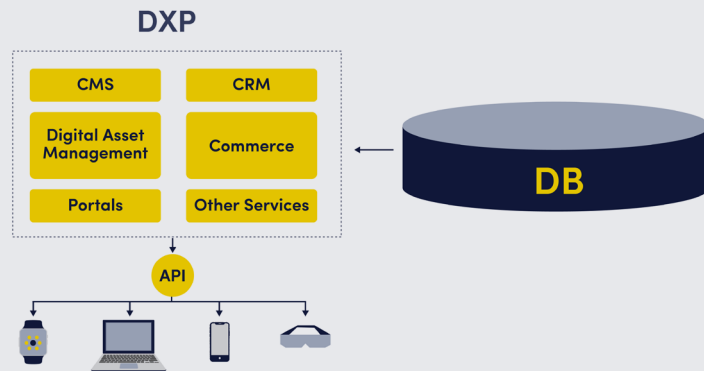


Decoupled CMS



- Headless and decoupled CMS both allow decoupling of frontend and backend layers.
- Headless and decoupled CMS both allow content reuse.
- Headless CMS is backend-only, whereas decoupled CMS includes an optional frontend.
- Decoupled CMSs often offer templates and themes out-of-the-box, whereas headless CMSs don't.
- Headless CMS gives developers maximum flexibility over the frontend technologies they want to use. With decoupled CMS, their flexibility is slightly limited, but in turn, content creators and marketers get to use tools like template management, content previews, and WYSIWYG authoring.

Headless CMS vs. DXP platforms

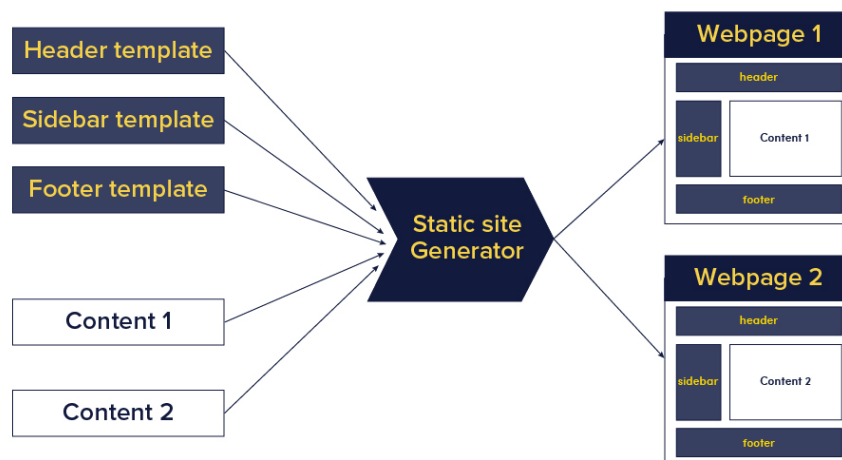


- DXP (digital experience platform) is a suite of tools and services that enable companies to produce tailored content for their customers. On the other hand, a headless CMS is an application that offers an efficient way of managing and delivering content.
- A CMS (can be headless) is often an integral part of the DXP tool suite.
- Both DXP and CMS enable you to implement an omnichannel content delivery strategy, but a DXP usually offers additional features, like data analytics and customer experience personalization.
- A DXP is much costlier than a headless CMS.

Headless CMS vs. Static Site Generators (SSGs)

- SSGs or static site generators use static HTML files to display content to the user, whereas headless CMSs typically use dynamic rendering.
- A headless CMS can act as a data source for an SSG.
- SSG websites are easier to build, but lack the feature-richness of dynamic websites powered by headless CMSs.
- SSGs can only be used for websites, whereas a headless CMS can help you achieve omnichannel content delivery.
- SSGs don't require server-side processing, whereas headless CMSs do.
- Headless CMSs offer a consolidated content creation experience; SSGs don't.

Static Site Generator



Why would someone need a headless CMS?

You should go for a headless CMS if you want:

- Segregation of your frontend tech stack from your backend, for better code management and less technical debt.
- Faster delivery of the same content to multiple channels, devices, and touchpoints.
- Maximum compatibility with all the frontend frameworks and technologies.
- To scale quickly and cater to an ever-growing visitor base.
- To deliver personalized user experiences, using data captured from different customer touchpoints.
- To have a future-proof technology stack that will seamlessly integrate with any upcoming digital touchpoints.

Pros of going headless

- **Faster editing:** As a headless CMS doesn't have any rendering overhead, it affords a fast and smooth editing experience.
- **No technology lock-in:** Different development teams can choose different front-end technologies (Angular, Vue.js, React etc.).
- **Faster time to market:** A headless CMS enables you to quickly put together an MVP, decreasing your time to market.
- **Reduce technical debt:** Instead of building a hard-to-manage monolithic application, use microservices that are more maintainable and scalable.
- **Go API-driven:** APIs have been the backbone of modern infrastructure for the last decade or so. They enable you to increase your application's functionality, as well as interoperability. Going headless allows you to harness the full potential of APIs.
- **Modernize your tech stack:** Establish yourself as an agile company by using the latest frontend technologies, and delivering content across multiple channels.

Cons of going headless

- Higher development costs: The segregation of frontend and backend stacks may lead to higher development and maintenance costs.
- They can get complex: Too much flexibility can be a curse for inexperienced or understaffed development teams. Coding the frontend from scratch, and managing the code for different front-ends can sometimes get complicated.

Headless CMS use cases

Headless enables you to serve your customers where they want. It's an excellent choice for building several kinds of websites and products. Let's look at a few:

E-commerce websites:

Modern e-commerce platforms, like Shopify, give you the luxury of implementing headless commerce. This approach allows you to use the stable, feature-rich backend of an e-commerce platform with a frontend of choice. Headless commerce suits people who want to deliver personalized shopping experiences.



Scalable, feature-rich websites:

With headless, you have the best of both worlds: the ease of content management that a CMS offers, and the feature-richness and customizability of modern frontend frameworks, like dynamic routing, client-side navigation, fast refresh, and lazy loading components etc.

Digital products and services:

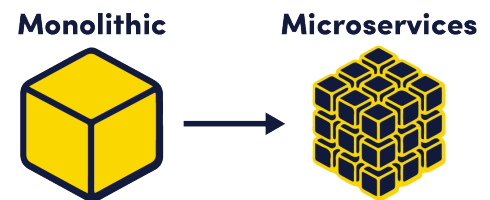
A headless CMS empowers you to go beyond websites. From kiosks to smart devices and conversational interfaces, you can use it to serve content on any number of digital products. This allows you to deliver rich and engaging omnichannel experiences.

Native mobile applications:

You can easily build mobile applications (iOS, Android, Windows) that fetch content from a headless CMS using its APIs. This allows you to avoid the development, maintenance, and operational costs of setting up a separate mobile backend to serve the same content.

Microservices:

Headless CMS fits perfectly into a microservices architecture. You can build loosely-coupled microservices for different use-cases and serve content to all of them via the APIs exposed by a headless microservice. E.g. you may create an auto-scaling Kubernetes cluster in which multiple headless CMS pods deliver content to different frontend pods.



Considerations before looking for a CMS

There are several factors to consider before you begin your search for the perfect headless CMS. You must carefully document your requirements, discuss ownership, and establish your problems and goals.

SEO

Your headless CMS should be your biggest SEO-enabler. It should have built-in features to optimize content for different devices, increase page load speed, create metadata, and **add structured content**.

Know your team

Start with your team. How many developers do you have? Do they have any experience with API development, or with the frontend stack you plan to use after going headless? Will they need training to get started? It's important to ensure that the chosen CMS doesn't have a steep learning curve.

Moreover, how do you currently produce, manage and maintain content? Do your marketers and content creators require developer assistance to ship out content? How do you expect the process to change after going headless? Will they be ready to trade-off some self-reliance for a much better content delivery mechanism?

It's crucial to ensure that everyone is on the same page, and receptive to change, before you begin looking for a headless CMS.

You must also define your primary stakeholders and their requirements. For example, which team/stakeholder will own the headless transition/adoption? If you are migrating, how much time are you setting aside for the migration, and who will lead the data migration process? You may need to assign project managers, technical architects, or seek professional third-party help.

Know your requirements

Are you buying a headless CMS to start your implementation from scratch, or will you be migrating from a legacy CMS? Do you plan on incremental migration, or at once? If it's the latter, how do you plan on automating the data migration? Moreover, is the legacy CMS integrated with other software? Typically, you should be looking for a highly interoperable headless CMS which supports the same integrations, and more.

It's also important to decide how you will be evaluating potential CMSs. Will you ask for proof-of-concepts or will a demo suffice? For how long would you like to try out the demo, and who would have access to it?

What are your technical must-haves, nice-to-haves, and deal-breakers? E.g. you may want to have scheduled publishing, version control, and the ability to set permissions and roles. Not having localization support may be a deal-breaker for you.

While deciding on your desired feature list, ensure that you only include the features that you will use, or could use in the future. By extension, don't go for a headless CMS, simply because it has the most features. Think about the usefulness of the offered features, and be sure not to mistake feature bloat for feature richness.

Know your problems and goals

What problems are you planning to solve with a headless CMS? It's important to carefully validate your need for a headless CMS. This will help you gain all the necessary approvals for budgeting and procurement.

Create a list of goals that you want to achieve. E.g. you may want to implement an omnichannel strategy, you may want to deliver tailored customer experiences, or you may want to integrate with an e-commerce platform. At this stage, imagining your ideal content management setup should help you come up with an exhaustive set of goals.

Considerations when picking a CMS

Now that we have done the groundwork, we are ready to start our search for a headless CMS. To make the right choice, you need to consider and evaluate several things.

API support

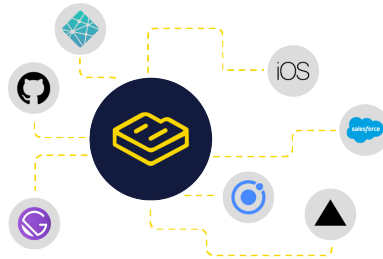
If there's one thing you absolutely can't compromise on, it's the API architecture. It should be easy-to-use, well-documented, and scalable. It should support all the necessary operations like ordering, filtering, server-side searching, pagination, and deletion. If you need to customize the API according to your business needs, ensure that the vendor supports it.

Multilingual functionality

You should be able to add multiple languages to your website, directly from the configuration panel. People love reading in their native language, and if your CMS-of-choice can **localize the content** based on the visitor's location, that will significantly enhance the user experience.

Integrations

The headless CMS should allow seamless integrations with third-party applications, like CRMs and marketing tools. There should also either be a marketplace for ready-to-install integrations or provide tools like a write API to allow you the flexibility to build the custom support with other applications that you need..



Roles and permissions

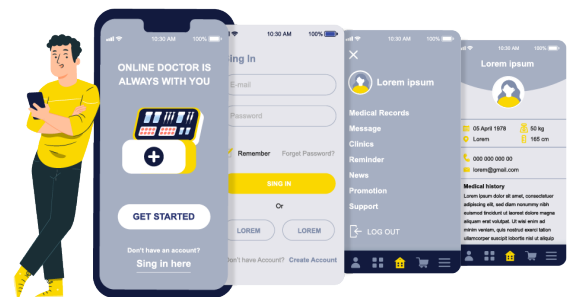
Ensure that the administrative panel supports different roles, like authors, publishers, and admins. The interface should allow you to specify the permissions you want to associate with each user. This will be key in ensuring that nobody has more permissions than they need.

Content editor

Everything comes down to content. The editor should be intuitive, responsive, and user-friendly. It should have features like version control, scheduled publishing, workflows, previewing and more. The ability for any business user to create new content and update existing content should come with easy due to search and filter capabilities within the editor UIs and dashboards.

User interface

The usability of your CMS should be high for all user types, including developers, administrators, and content creators. E.g. Developers should be able to test APIs using the tool, administrators should be able to assign roles, approve and schedule content directly from the UI, and content creators should have a feature-rich editor that offers quick previews.



Pricing

Pricing can be a highly subjective matter, but ideally, the vendor should have a flexible pricing model that gives you multiple packages to choose from, with an option to upgrade as per scaling needs. If no package fits, you should have the option to request a customized plan. Moreover, keep an eye out for any hidden costs like overage fees and setup or maintenance charges.

CDN + Asset management

If you plan on using a CDN to deliver your content, ensure that the headless CMS supports working with it or has its own CDN so you potentially no longer need to manage one. Implementing middle-layer caching using a distributed CDN will provide you with speedy content delivery at a global scale.

The headless CMS should have built-in support for creating, saving, and managing digital assets, or the ability to integrate with a digital asset management (DAM) tool.

Versioning

Version control will enable your editors and authors to keep track of all the changes made to a piece, and revert to an older version if and when needed.

Multi-site support

If you have multiple websites, or plan to have in the future, choose a CMS that lets you manage all of them from a single platform. This will allow you to reuse the same content and code, publish content quickly, and reduce administrative costs.



Security

Cyberattacks are happening at an alarming rate. It's crucial to take security seriously. Choose a CMS that lets you keep your content and data safe. Top security considerations include API security, ability to blacklist/whitelist IPs, detect malicious traffic, and regular vulnerability scanning and patching.

Modern SaaS CMSs, like ButterCMS, handle all aspects of security for you, and you don't have to worry about manually installing updates or patches.

Customization

How easy/hard is it to customize the CMS? Do you have any specific customization needs, and can the vendor meet them? E.g. you may need a new API endpoint to fetch paginated content, or you may want to request an integration with your own custom APIs.

Hosting: On-premise or as a service

Do you want the CMS to be hosted in-house, or delivered as a service? Even though most people prefer the SaaS model, you may have different compliance or regulatory requirements. Engage with the relevant stakeholders, and examine your business' needs and preferences to make the right call.

Support

You should have multiple support options, including a customer success manager, live chat support, and excellent documentation. Your developers and admin team should know where or who to look for, in case they hit a snag.

If you are considering an open-source solution, ensure that there is a well-organized developer community that is actively supporting it. You may get an idea by the number of open issues on GitHub, and the speed/accuracy with which any previous bugs/vulnerabilities were solved. The availability of optional paid support should also be checked.

Data storage and maintenance

Your content and customer data should be automatically backed up across different geographical locations. The vendor should have ISO-27001 certified data centers and third-party providers. Moreover, they should also be compliant with various data protection laws, like GDPR, CCPA, and DPA 2018 etc.

Questions to ask when considering a headless CMS

To further contextualize your search for the right headless CMS, ask the following questions:

Developer related questions

- What programming languages is the CMS compatible with? Or is it language-agnostic?
- What API protocol/architecture does it have? Is it REST/SOAP?
- What programming languages is the CMS compatible with? Or is it language-agnostic?
- What API protocol/architecture does it have? Is it REST/SOAP?
- Is there built-in support for different environments, e.g. developer, staging, and production.
- Is the CMS extensible? i.e. is it open-source or closed-source.
- Is there a write API that developers can use to write integration logic for third-party services and tools?

Marketing related questions

- How do my other marketing tools work with the CMS?
- In what ways does the CMS help to create, organize, publish rich media assets, such as images and videos?
- What are the limitations for creating new content as a non-technical business user?
- Does it make content creators more autonomous, effective and productive in their jobs?
- If migrating from a legacy CMS, how big of a learning curve will it have and how with the CMS support our team to be successful?

Business related questions

- What is the estimated cost and timeframe for implementation?
- Are there any migration costs involved?
- Is it agile enough to meet future demands?
- Will there be any disruption or downtime?
- How security-conscious is the vendor? Have they been breached before?

Headless CMS evaluation checklist

To sum it all up, here's a condensed evaluation checklist that summarizes everything you should look for in a headless CMS:

CMS Architecture

- Robust, scalable, and well-documented APIs.
- Has a high usability score.
- Version control that can take you back in time.
- SaaS or on-premise, based on business needs.

Implementation

- Supports utilization of third party applications.
- Support for multiple languages and localization.
- Ability to set roles and permissions.
- Dedicated, professional support.
- Customizability as per your business needs.

Content Management & Editorial Features

- Out-of-the-box SEO enablement.
- An intuitive, responsive content editor.
- Native asset management, or integration with a DAM tool.
- Ability to manage multiple sites from the same platform.

Business & Security Considerations

- Flexible pricing.
- Rigorous security and compliance with industry standards.
- Automated periodic backups and compliance with data protection laws.

Headless CMS usage in real life

Headless CMSs enable businesses to scale, publish content effectively, and deliver optimized customer experiences. Let's look at a few real-life examples:

How ButterCMS catalyzed PickFu's website redesign by decoupling its marketing and developer workflows

PickFu is an online platform for consumer research. Up until recently, they had been using a hard-coded website, which meant that changes on their website needed to be done manually by a developer. This created unnecessary friction between the marketing and development teams, along with a bottleneck in their content creation process.

The solution was simple: Redesign the website such that the marketing and development teams are decoupled, and made self-reliant. Allow developers to focus on development tasks, and enable marketers to efficiently make changes to the website.

The PickFu team quickly realized that a headless CMS, like ButterCMS, would solve all their problems. They integrated with ButterCMS seamlessly, and used its components feature to give their marketers full control over creating personalized pages, without developer assistance.



By decoupling marketing and dev, and streamlining the structuring, creation, and publishing of pages, they have been able to double pages-per-visit on their website, and deliver a much-enhanced customer experience.

How Freddie's Flowers used ButterCMS to scale internationally

Freddie's Flowers is a flower delivery service based out of the UK. As Freddie's went international, their legacy, static website couldn't keep pace. Marketers couldn't perform rapid A/B testing on their content, without requiring developer assistance. This led to weeks-long waits to push out changes. It was soon established that a more efficient way of delivering content was needed.

After doing some research, the team stumbled upon ButterCMS, a headless CMS that would forever change their content generation and delivery. It took 3-4 months for the developers to build their first headless-powered website, using Vue.js, PHP, and ButterCMS.

They were able to fetch all the required data using APIs, and customize the way it was rendered on the client's screen. The transition allowed Freddie's to build dynamic landing pages, support different locales, set up A/B tests, and update pages without requiring developer assistance.

How Realty Austin Used ButterCMS to Seamlessly Scale their Content Management

With a goal of becoming a premier local real estate agency, Realty Austin knew they'd be going up against some heavy industry hitters in the Google SERPs and would need a solid SEO strategy.

They also knew that they'd need a CMS platform that could grow with them, something they'd been struggling with after implementing new features in an effort to improve their customer journey and generate new leads.

After deciding to adopt a microservices approach to their architecture, it was only a matter of time before they discovered the power of a headless CMS. Once they began interacting with the team at ButterCMS, they were sold on its reliability, scalability, and reusability—and were able to get their MVP up and running in less than three months.



About ButterCMS

ButterCMS is a headless CMS that empowers businesses to effortlessly build, deliver, and optimize multi-channel, user-friendly experiences. With a few lines of code, you can integrate with Butter using any frontend tech stack, like Vue.js, PHP, Next.js, Flutter, and React etc.

Butter is powering the digital experiences of hundreds of companies:

Butter is powering the digital experiences of hundreds of companies:

tropical CAFE
SMOOTHIE



hims & hers 1800 contacts®

KAPLAN

indeed

NEWAGE



KAVAK



Whether you want to serve content to several digital products, manage multiple sites via a single platform, build a marketing website for your e-commerce business, or simply launch a blog; Butter is the ideal fit for virtually any use-case.

It enables you to scale at will, revitalizes your content management, empowers your marketers to publish content without developer assistance, and rids you of all the traditional CMS headaches. Oh, and did we mention it is a top rated headless CMS Leader on G2?

Try Butter free for 30 days, or view the **extensive docs** with SDKs and starter apps to get you up and running in no time. Or you can read how our customers are **solving real-life challenges using Butter**.

To learn more about how ButterCMS can fix all your burning content management problems, **get in touch with our sales team here**.

Experience Butter now.
30 day free trial. You'll be set up less.

Try it Free