**What Are Single Page Applications and Why Do People Like Them So Much?**

Single page applications (SPAs) are everywhere. Even if you’re not exactly sure what they are, you most likely use them regularly — they’re a great tool for making incredibly engaging and unique experiences for website users.

A single page application is a website or web application that dynamically rewrites a current web page with new data from the web server, instead of the default method of a web browser loading entire new pages.

You’ll easily recognize some popular examples of single page applications like Gmail, Google Maps, Airbnb, Netflix, Pinterest, Paypal, and many more. Companies all over the internet are using SPAs to build a fluid, scalable experience.

However, in the past, SPAs have left marketers in the dark when it comes to managing content. Working with a single page application has historically been a code-heavy undertaking that doesn’t cater to a marketer’s wants or needs. Luckily, it’s now possible to pair your SPA with [the right content management system (CMS)](https://www.bloomreach.com/en/blog/content-management-system)to give both developers and marketers the level of control they are looking for.

Ready to learn all about SPAs and what they can do for your e-commerce business? Here’s everything you need to know.

Key Takeaways

Learn more about how a single page application (SPA) can help you even further succeed with your content management system.

1. An SPA is a website or web application that dynamically rewrites a current web page with new data from the web server, instead of the default method of a web browser loading entire new pages.
2. SPAs offer a fluid and dynamic user experience by loading all required resources during the initial page load. As users interact with the application, the content is updated dynamically, eliminating the need for page reloads and reducing server requests.
3. Improved application performance, consistency, reduced development time, and lower infrastructure costs are all key advantages of SPAs.

**What Is a Single Page Application?**

A single page application is a single page (hence the name) where a lot of information stays the same and only a few pieces need to be updated at a time.

For example, when you browse through your email, you’ll notice that not much changes during navigation — the sidebar and header remain untouched as you go through your inbox.

The SPA only sends what you need with each click, and your browser renders that information. This is different than a traditional page load where the server rerenders a full page with every click you make and sends it to your browser.

This piece-by-piece, client-side method makes load times much faster for users. It also lessens the amount of information a server has to send and makes the whole process a lot more cost-efficient — a win-win scenario for users and businesses.

**What Is Single Page Application Architecture? How Does It Work?**

It’s obvious to see the benefits of a single page application. Utilizing a web application or website that interacts with the user by dynamically rewriting the current page, rather than loading entire new pages from the server, provides a much better user experience.

It avoids interruptions in a user’s journey, which is vital for websites — especially in [digital commerce](https://www.bloomreach.com/en/blog/digital-commerce-explained). By cutting down on lag time between successive pages, it makes the site behave more like a desktop application, providing a more fluid and comfortable experience.

And this can make a huge impact. Because on most websites, there is a lot of repeating content.

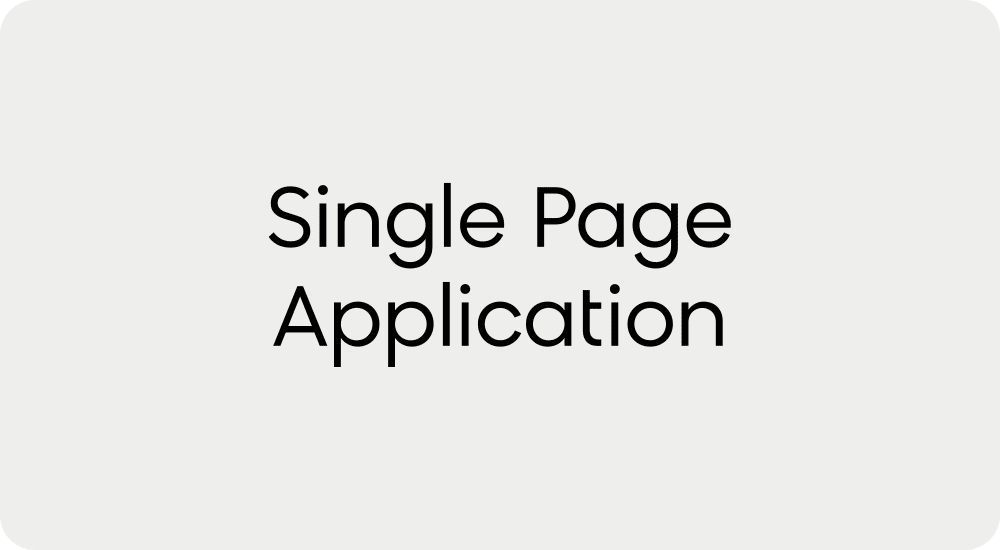
Some of it stays the same no matter where the user goes (headers, footers, logos, and the navigation bar), while some of it is constant in just a certain section (filter bars and banners). And there are many repeating layouts and templates (blogs, self-service pages, or the aforementioned Google mail setup).

Single page applications take advantage of this repetition.

Let’s say you visit a website and view a painting of a house and a tree. Traditional multi-page websites paint the entire picture for you on the server and send it over to your browser.



An SPA gives you the paint-by-numbers guide for the site, including the repeating guides you’ll likely be using, and then pipes in the right paint (data and content) to fill out the template.



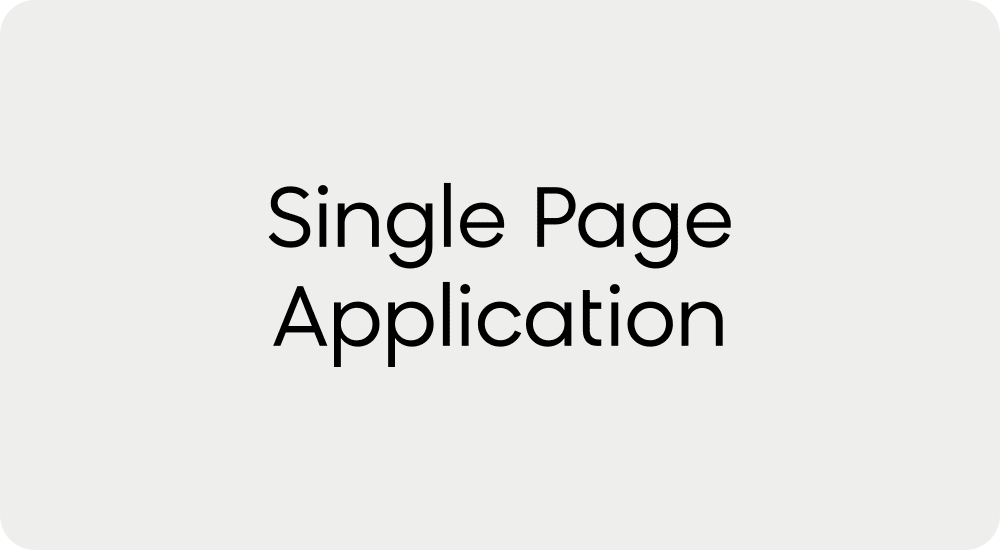
Either way, you see the same tree. But the speed of single page applications comes in when you request new content — like clicking on “next,” filtering results, opening an email, or in our case, asking to see a different tree.

On a traditional website, your request for a new tree would cause the server to repaint the entire picture and send it back.



With a single page application, the server says, “Hey, I’ve got a new tree for you, but you’ve already got the house so just leave that the same.” Then, it sends the page updated instructions for a new tree and the paint to make it.

By transferring the painting work (or page rendering) from the server to the client (you), the page can be dynamically rewritten instead of going through an entire reload. This makes things a whole lot faster.



**Advantages of Single Page Applications**

There are many benefits to SPA solutions on both the [customer experience](https://www.bloomreach.com/en/blog/ecommerce-customer-experience) side and back-end design side of the equation. Improved application performance, consistency, reduced development time, and lower infrastructure costs not only help you offer a more pleasant experience to your users, they also help development teams operate more effectively.

By separating the presentation from the content and data, development teams can work at different speeds while still remaining integrated and working towards the overall solution. SPAs are also good for making responsive designs for mobile, desktop, and tablet.

**[Advantage #1] A Single-time File Load Each For HTML, CSS, JS**

With a single page application, after the initial page load, the server doesn’t send any more HTML to you —  you download it all right at the beginning. The server sends you a shell page and your browser renders the user interface (UI).

Then, as you click around, the SPA sends back requests for data and markup. The server shoots back the raw materials needed, and your browser takes it and renders an updated UI, interchanging pieces without ever needing to refresh the full page. This quick interchangeability makes SPAs incredibly useful on pages that are highly navigated and use repeating templates.

**[Advantage #2] No Extra Queries to Server**

Because the server doesn’t need to spend time and energy doing multiple renderings, SPAs lower the impact on your servers overall, meaning you can save money by using fewer servers for the same amount of traffic.

**[Advantage #3]  Fast and Responsive Front-end Building**

Along with a quicker performance time, SPAs also let developers build the front end of sites faster. This is due to the decoupled architecture of SPAs, or a separation of back-end services and front-end display.

Many business-critical functionalities don’t change all that much on the back end. How your customers log in, register, purchase, and track orders may change its “look” or presentation from time to time, but the logic and data orchestration behind it is pretty constant — and you don’t want to risk messing that up.

Similarly, your raw content and data might stay the same, but the way you want to display it may differ. By decoupling that back-end logic and data from how it’s presented, you turn it into a “service,” and developers can build many different front-end ways to show and use that service.

With a [decoupled setup](https://www.bloomreach.com/en/blog/2018/01/what-is-digital-experience-platform-dxp), developers can build, deploy, and experiment with the front end completely independent from the underlying back-end technology. They design how they want the user experience to look and feel, and then pull in the content, data, and functionality through those services.

This is done using APIs, which are a standard set of rules between applications on how they will structure, exchange, and reassemble data.

This API setup lets developers work quickly on the UI with no risk to business-critical back-end technologies.

Read this next: [What Is Composable Commerce? Why Does It Matter?](https://www.bloomreach.com/en/blog/composable-commerce) [blog]

**[Advantage #4] Enhanced User Experiences**

As more and more functionalities are built as modular services (a microservice architecture) that can be updated independently, it becomes easier to experiment with how they are displayed and used.

[SPA frameworks](https://www.bloomreach.com/site/en/blog/2018/personalization-and-hybrid-support-for-single-page-applications) are great for trialing these services to create engaging, dynamic, and even animated user experiences.

Plus, many people simply prefer developing in a certain programming language (many SPA frameworks use JavaScript) and thanks to APIs, the SPAs you build in one language can work seamlessly with back-end services developed in different languages.

**Read this next:**[**What Is Omnichannel Commerce? Definition, Benefits, and Trends**](https://www.bloomreach.com/en/blog/omnichannel-commerce-for-business)**[blog]**

**Angular vs. React vs. Ember vs. Vue — Which Framework Is Best for Single Page Applications?**

Angular, React, and many others (such as Ember and Vue) are frameworks that developers use to create SPAs efficiently and eloquently.

Simply put, these frameworks are a collection of reusable components, which many developers have contributed to, that follow a defined set of building rules.

If you equate building an SPA to building a house, you could mix up the clay, dry the bricks, and mine and mold the steel yourself — or you could use the bricks and pipes other people have already designed and focus your time on what makes your house unique.

[There are differences between them all](https://medium.com/unicorn-supplies/angular-vs-react-vs-vue-a-2017-comparison-c5c52d620176), but a great thing about SPAs and the frameworks that support them is that, thanks to APIs and the right integrations, you can use whichever framework you prefer with your other API-enabled technologies.

**Why Single Page Applications and Content Management Systems Are Historically a Difficult Pairing**

When using SPAs, developers may think of the experience as an “app.” But a website visitor is still going to think of it as a webpage. And where there is a webpage, there is a marketing team itching to optimize it.

Because SPAs are apps that require development work to tinker with the display and delivery of the experience, marketers have to ask developers for help with every tweak they want to make — causing inevitable bottlenecks.

**[Reason #1] An Absence of Editing Tools That Marketers Understand**

The usual editing features found in content management systems that marketing teams rely on, like live preview, drag-and-drop, and WYSIWYG editing, are usually tied to the delivery tier in a CMS.

But it gets tricky with SPAs. The delivery tier is determined by the SPA and the content is simply stored in the CMS in a standard way that the APIs can read. Because the SPA is rendered on the front end, the back-end CMS has no idea what it should look like and therefore can’t spin up a preview.

So CMS users end up getting stuck with a very dated approach — fill in a form, cross your fingers, push publish, and go see what it looks like live.

To return to our paint-by-numbers example, the CMS stores the raw content (the paint) and the SPA has the paint-by-numbers guide of how that content should look. The preview doesn’t have this guide, so it can’t figure out what the content should look like.

This is a pure “headless” delivery of content (it doesn’t have the CMS delivery tier as a “head”). It’s great for developing quickly, but is a bit rough for marketers who want to change the website on their own without having to code.

Additionally, marketers are used to thinking of site content in terms of “pages.” But because an SPA is a single page, the page-building and editing features marketers need aren’t available. If they want a new “page” (a “route” in an SPA), or want the view to look differently, they have to ask a developer to make those changes.

**Read this next:**[**Headless CMS vs. Decoupled CMS: The Complete Guide**](https://www.bloomreach.com/en/blog/2019/05/what-is-customer-experience)**[blog]**

**[Reason #2] Difficulties in Reusing Content**

This issue arises from two core reasons: antiquated content management systems and SPA design.

First, there are certain content management systems where there simply isn’t a decoupling of how your content looks and how it’s stored. Because the storage of content is not in a standard, presentation-neutral format, the SPA can’t use it in the API-based way it wants to.

This isn’t just a problem when using SPAs, of course. This kind of CMS setup makes it impossible to reuse content across channels in general. Because the content is tied to how it’s displayed (a page-based system), the FAQs you put on your website can’t just be tapped for someone to flip through on their smart watch — you’d have to store the same content in two different ways.

SPAs need a content-based CMS to work properly so that it can pull raw content and display it however it needs to be presented.

On the SPA side, the difficulty comes from the fact that many websites are now designed for a hybrid setup.

While you may want some parts as an SPA, you might want others set up in the traditional mutli-page site format, which is often better for SEO. Despite the two different approaches, there needs to be a cohesive feel between these pages.

If your setup is two buckets of content (pieces for the traditional site and the SPA), that cohesion is going to break. You need content that works across both modes to present a seamless user experience.

**[Reason #3] Difficulties With Personalization and Relevance**

There is an ever-increasing need for websites to provide [personalized experiences](https://www.bloomreach.com/en/blog/2019/07/digital-commerce-explained) for users, especially for e-commerce businesses. Without the neccessary architecture in place to handle a sophisticated level of personalization, SPAs can fall tragically short.

The problem is, personalization and relevance capabilities traditionally rely on complete page loads to render experiences. They require [real-time data](https://www.bloomreach.com/en/products/data-engine), and without the right CMS, SPAs can’t display content with that much context. This can be a problem when your goal is relevant and [personalized content](https://www.bloomreach.com/en/products/content/content-personalization) delivery.

On top of that, a number of content management systems implement personalization on the client side. Unfortunately, those JavaScript personalization rules don’t play very nicely on top of the SPA javascript.

Read this next: [The Personalization Puzzle: Assembling the Pieces of Connected Commerce Experiences](https://www.bloomreach.com/en/blog/the-personalization-puzzle-assembling-the-pieces-of-connected-commerce-experiences) [blog]

**Are Marketers Doomed in an SPA World?**

Of course not! You simply need a CMS that has the right architecture in place for SPA use — one that is API-based and decouples content from presentation. And you need a system that can work with an SPA to provide editing tools and live previews, support a hybrid setup, and allow for personalization on the server side.

That is exactly what [Bloomreach Content](https://www.bloomreach.com/en/products/content" \t "_blank) offers. Our [headless content](https://www.bloomreach.com/en/products/content/headless-cms) platform is built for e-commere, allowing you to easy manage your site and optimize every user touchpoint with Bloomreach’s powerful APIs and software solutions designed to help you capture more revenue.

And personalization has never been easier when you combine the power of our headless experience manager with [Bloomreach Engagement](https://www.bloomreach.com/en/products/engagement" \t "_blank), our real-time customer [data platform](https://www.bloomreach.com/en/blog/customer-data-platform) with intelligent marketing automation.

Want to learn more about how Bloomreach can help your business? Read all about our game-changing technology:

* [How Bloomreach Offers Single Page Application Ease for Developers and Marketers](https://www.bloomreach.com/en/blog/2018/08/cms-that-works-with-spas-angularjs-react)
* [Personalization and Hybrid Support for Single Page Applications](https://www.bloomreach.com/en/blog/2018/08/personalization-and-hybrid-support-for-single-page-applications)

If you’re ready to implement a content management system for your company’s future, [schedule a personalized demo with Bloomreach today](https://www.bloomreach.com/en/request-demo).