

# Monetate and SEO

Search engine optimization (SEO) is the process of getting pages to rank higher in organic (non-paid) search engine results on various search engines like Google, Bing, or Yahoo. SEO encompasses technical considerations such as load time, keywords, and creative elements and content to improve rankings, drive traffic, and increase awareness in search engines.

Search engines use automated crawlers (bots) to read sites and then parse the pages, interpret the results, and index and cache the content. This content is then analyzed via various proprietary algorithms to generate a ranking system used with search keywords to show appropriate links to users.

## Page Speed

Google has stated in [Using site speed in web search ranking](#) that site speed and, as a result, page speed is one of the signals used by its algorithm to rank pages. Research cited in [How Website Speed Actually Impacts Search Ranking](#) has shown that Google might be specifically measuring time to first byte when it considers page speed. In addition, a slow page speed means that search engines can crawl fewer pages using their allocated crawl budget and this could negatively affect your index.

Everything about Monetate's code, the Monetate tag, and the content it serves on your site is designed to be as quick as possible. Monetate's engineers take numerous steps when developing actions, experiences, and the Monetate platform itself to ensure content is loaded quickly and seamlessly for users across multiple platforms.

Monetate only uses the Sizzle library for selectors. Everything else is built with vanilla JavaScript.

## Algorithms Optimized for Mobile

In recent years, search engine algorithms have grown more sophisticated and require access to more content, including JavaScript and CSS files. Search engines such as Google need to render the complete webpage to ensure the best experience for the user.

Google wants to ensure pages are mobile-friendly to apply both the mobile-friendly tag in the search results and the associated ranking boost for mobile search results.

## Algorithms Focused on User-Friendly Page Layout

Part of Google's page layout algorithm looks at where content is placed on the page in relation to the advertisements. Through the questionable use of CSS or JavaScript, Web developers can easily make it appear that the content is front and center while the ads are the most visible part of the page above the fold. This is done in an attempt to artificially prop up their search ranking while increasing the drive to advertisements.

If Google's algorithm determines a webpage is mostly ads above the fold with the actual content below the fold, it can devalue the rankings for those pages.

If a site denies search engine robots access to their CSS and JavaScript files in an attempt to prevent this algorithm from impacting them, they may be penalized for other reasons. Google states, "Disallowing crawling of JavaScript or CSS files in your site's `robots.txt` directly harms how well our algorithms render and index your content and can result in suboptimal rankings."

## Impact of Personalization and Testing Impact

Google and other major search engines don't discourage A/B testing or personalization. Read more about it in [Website testing and Google search](#).

Google does its utmost to avoid inadvertently penalizing search rankings of sites that employ testing and optimization. To this end, it provides several key pointers to ensure your testing remains as effective as possible and ultimately provides the best user experience.

### No Cloaking

Cloaking—showing one set of content to humans and a different set to Googlebot—is against Google's Webmaster Guidelines, whether you're running a test or not. Ensure that you're not deciding whether to serve the test or which content variant to serve based on user-agent.

Always serving the original content when you see the user-agent `Googlebot` is an example. Remember that infringing Google's guidelines can get your site demoted or removed from Google search results—probably not the desired outcome of your test.

### Use `rel="canonical"`

If you run an A/B test with multiple URLs, you can use the `rel="canonical"` link attribute on all your alternate URLs to indicate that the original URL is the preferred version. Monetate recommends using `rel="canonical"` rather than a `noindex` metatag because it more closely matches your intent in this situation.

If you're testing variations of your homepage, you don't want search engines to avoid indexing your homepage. You want them to understand that all the test URLs are close duplicates or variations on the original URL and should be grouped as such with the original URL as the canonical. Using `noindex` rather than `rel="canonical"` can sometimes have unexpected effects.

Consider the following example. If for some reason you choose one of the variant URLs as the canonical, the "original" URL might also get dropped from the index since it would get treated as a duplicate.

### Use 302s, not 301s

If you run an A/B test that redirects customers from the original URL to a variation URL, use a 302 (temporary) redirect not a 301 (permanent) redirect. This tells search engines that this redirect is temporary—it only remains in place as long as you're running the experiment—and that they should keep the original URL in their index rather than replacing it with the target of the redirect (the test page). JavaScript-based redirects are also fine.

### Run Experience Only as Long as Necessary

The amount of time required for a reliable test varies depending on factors, such as conversion rates and how much traffic your site gets. A good testing tool should tell you when you've gathered enough data to draw a reliable conclusion.

Once you've concluded the test, update your site with the desired content variation(s) and remove all elements of the test as soon as possible. These testing elements may include alternate URLs or testing scripts and markup. If Google discovers a site running an experiment for an unnecessarily long time, it may interpret this as an attempt to deceive search engines and take action accordingly. This is especially true if you serve one content variant to a large percentage of your users.

## Full-Page Testing Implications

Full-Page Test (FPT) experiences are different. In an FPT experience, Monetate considers many factors before it decides whether to keep a customer on the base page or send them to a completely different URL. You can target those tests, and Monetate doesn't want you to lose that ability. This means that an FPT experience must be done in JavaScript once the browser has requested and seen the base content. In other words, an FPT experience can't be done using a server redirect (a 302 Temporary redirect).

Furthermore, Google appears to treat those as a permanent redirect (301).

FPT experiences, however, aren't a lost cause. You can give a hint to the search engines about what you're doing by ensuring the alternate page includes `rel canonical` information in the source code that points back to the original URL. This tells the search engines that you are not trying to hide different content under a page, but rather that the page that is specified is the canonical source and is the one that should be indexed.

Unfortunately, this is not something Monetate can do for you. Monetate works on the front end and modifies the page after it's served. This information sits in the `<head>` of the page and needs to be there when the page is served by your servers.

For experiences that redirect some customers to another page, it may also be very useful to add `rel=nofollow` to the HTML that you insert to tell search engines not to follow the link and index the next page. Where possible, you should avoid running Full-Page Test experiences for too long because this can also have an adverse effect on the way that Google measures page ranking.

## SEO Bot Engagement with Experiences

When you add the Monetate tag to your site and activate experiences for testing and personalization, the tag (which is JavaScript code) is fetched from Monetate's servers. That code collects information and sends a request once the browser has parsed the page. Monetate systems then determine what experiences to run for this given page view.

In the past, Monetate stopped this process from occurring when the user agent (or browser fingerprint) was identified as a bot. Due to the increasing sophistication in search engine algorithms and how they gather and interpret data, many search engines can readily determine if their bots are served content that differs from what a human user sees.

To account for this evolution in the search engine industry, Monetate has made several alterations and

improvements to its infrastructure so you can continue to test and personalize without jeopardizing your search engine ranking.

## What Monetate Is Doing to Help

Monetate does everything it can to ensure that it doesn't impact search engine results negatively, but your conversion goals, whatever they are, are our top priority. Full-page testing and personalization are critical pieces of your marketing efforts. They increase lift and conversion and generally let you serve your customers better.

If you leave experiences running for agility to make a quick change without dedicating your own development resources while your base site is updated with new content, bots see those changes and those updates factor into your search rank immediately. This makes it extremely beneficial to test potential re-platforming, content, or cosmetic changes to see if there could be any long-term impact on your search rank prior to implementing anything permanently to your site.

While in theory content served via experiences should be reflected in your search rank, Monetate has no direct control over how the content you serve is interpreted by search engine algorithms. If you run an experience that is interpreted as detrimental to the overall user experience, it impacts your search rank for that page accordingly.

- At clients' request, Monetate allows bots access content (including code and assets) previously hidden from them. Monetate configures those servers to ask search engines not to index and cache the assets as it understands those might be transitory on your site.
- Monetate no longer stops its features and functions from working for bots. They get the full experience, just like humans.
- Bot views aren't included in part of the analytics data you see in Monetate.

## Roadmap and Next Steps

With the numerous model-based features in the platform, Monetate converges on the right content for the right audience much more quickly and automatically. This means that it can serve crawlers the same content as humans. In the case of predictive testing, human or search engines quickly converge to one option.

## The Real Solution

Despite all this mitigation, crawlers that run different levels of JavaScript see different content. Monetate believes it has a minimal effect on rankings but may affect search results if a customer searches for content that's only served through Monetate experiences.

If you believe it's critical for crawlers to see the modified content because of the type and prominence of the experiences you run, then you can serve fully digested content to crawlers.

The first solution is to pre-render content for crawlers by recognizing the user agents and doing all of the work server-side. Some frameworks (such as [prerender.io](#)) do the heavy lifting for you. It's possible to configure your CDN endpoints to use different upstream servers for those user agents by pointing crawlers to pre-rendered content and the rest to dynamic content.

The second solution is to use the Monetate server-side integration, a feature that wires Monetate's decisioning in

your servers' pipeline before the page is sent back to the customer. Prominent long-running experiences can be configured through this integration point instead of the traditional Monetate tag implementation. This requires more work on your end since it's impossible for Monetate to provide out-of-the-box implementations in this scenario.