

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.