harper

Harper is a composable application platform that fuses database, cache, application, and messaging into a single, high-performance runtime.

Among many use cases, Harper provides prerendering for bots and users, accelerating SEO visibility while directly improving real-world speed, freshness, and resilience at the infrastructure layer.

PRERENDER

Prerender.io is a crawler-focused prerendering middleware. It detects bot user-agents (search, social, or AI crawlers) and returns cached HTML snapshots to those bots only. It enhances crawlability and indexation but does not improve human user performance, Core Web Vitals, or uptime.

Architectural Role

Functions as a distributed backend platform that can host entire applications.

Supports both bot and user delivery through edge-distributed architecture.

Operates at the infrastructure layer, replacing or augmenting legacy stacks.

Enables resilient fallback as cached pages are served when origin systems go down.

Acts as a middleware layer between web servers and crawlers.

Serves cached HTML to crawlers only, leaving human traffic to the origin.

Operates at the SEO middleware layer with no data-layer awareness.

Dependent on external uptime, it cannot serve users during outages

Ideal Use

Engineering-led organizations seeking full-stack performance gains.

High-growth e-commerce with dynamic data and large catalogs.

Companies that are consolidating technology or moving computation closer to the edge.

Use cases requiring data freshness, uptime, and speed for both bots and users

SEO or marketing teams needing quick crawl improvements.

Small to midsize sites running JavaScript-heavy front-ends.

Teams wanting plug-and-play SEO middleware.

Situations focused solely on indexation and crawl budget.

Core Offering

Unified runtime combining database, cache, app logic, and messaging into a high-performance runtime offering a broad set of capabilities.

Customizable TTLs and dynamic attribute injection to keep cached page data fresh even when longer TTLs are used.

Global edge delivery and event-driven freshness.

Resilient failover: keeps serving pages if the origin fails.

Composable performance layer extendable to APIs, caching, and messaging.

Standalone prerendering service for bots.

TTL-based re-caching updates crawler snapshots periodically.

Centralized render servers, adding latency.

Dependent on customer origin and CDN uptime.

Single-function SEO tool limited to prerendering.

Why Harper is a Better Enterprise Solution

Full Performance Impact, Bots and Humans

Harper improves both crawlability and user experience, directly enhancing Core Web Vitals and real-world interaction speeds.
Prerender.io improves only how bots see your site, not how users experience it.

Native Resilience and Edge Distribution

Harper's prerendered pages can serve even when origin systems fail, maintaining uptime and conversions for large catalogs.
Prerender.io depends on origin uptime and cannot serve as a failover layer.

Real-Time Freshness

Harper has TTL-based refresh cycles but can also dynamically inject updated data at request time without re-rendering, which is essential for maintaining live inventory, pricing, and personalization.
Prerender.io relies on TTL-based refresh cycles or manual re-render triggers.

Platform Value Compounds at Scale

Harper's unified runtime multiplies business impact as more workloads—data, cache, app logic—move onto the platform.
For large catalogs, Harper delivers exceptional value per render compared to prerender.io while also providing resilience as an origin backup.

Harper is built for enterprises that need fast, resilient backends to power both user and bot experiences, delivering performance gains and unmatched value as applications move closer to the edge.