

CHECKLIST TỐI ƯU FID (FIRST INPUT DELAY) WEBSITE

Biên soạn bởi: **HỒNG MINH (MINH HM)**

CEO LIGHT | Chuyên gia Marketing Online, lập trình với 12+ năm kinh nghiệm

HIỂU RÕ VỀ FID VÀ ĐO LƯỜNG

- ☐ Hiểu định nghĩa FID: thời gian trễ từ khi người dùng tương tác đầu tiên đến khi trình duyệt phản hồi
- ☐ Biết ngưỡng tốt của FID: $\leq 100\text{ms}$ (tốt), $100\text{-}300\text{ms}$ (cần cải thiện), $>300\text{ms}$ (kém)
- ☐ Phân biệt FID với các Core Web Vitals khác (LCP, CLS)
- ☐ Cài đặt Google PageSpeed Insights để đo FID
- ☐ Sử dụng Google Search Console theo dõi FID thực tế
- ☐ Cài đặt Lighthouse để kiểm tra FID định kỳ
- ☐ Theo dõi FID qua Real User Monitoring (RUM)
- ☐ Hiểu sự khác biệt giữa lab data và field data
- ☐ Thiết lập alerts khi FID vượt ngưỡng cho phép
- ☐ Ghi nhận các trang có FID cao nhất để ưu tiên tối ưu

TỐI ƯU JAVASCRIPT VÀ MAIN THREAD

- ☐ Phân tích và loại bỏ JavaScript không cần thiết
- ☐ Chia nhỏ JavaScript bundles (code splitting)
- ☐ Lazy load JavaScript chỉ khi cần thiết
- ☐ Sử dụng dynamic imports cho các modules không cần ngay
- ☐ Tối ưu thứ tự tải JavaScript (critical vs non-critical)
- ☐ Minify và compress tất cả JavaScript files
- ☐ Sử dụng Web Workers cho các tác vụ tính toán nặng
- ☐ Tránh long tasks ($>50\text{ms}$) block main thread
- ☐ Implement requestIdleCallback cho các tác vụ không urgent
- ☐ Debounce và throttle các event handlers

TỐI ƯU THỜI GIAN XỬ LÝ EVENT

- ☐ Tối ưu event listeners (click, scroll, touch, keyboard)
- ☐ Sử dụng passive event listeners khi có thể
- ☐ Tránh synchronous operations trong event handlers
- ☐ Ưu tiên xử lý input events quan trọng nhất trước
- ☐ Sử dụng event delegation thay vì nhiều listeners riêng lẻ
- ☐ Tối ưu DOM manipulation trong event handlers

- ☐ Cache DOM elements thay vì query lại nhiều lần
- ☐ Sử dụng DocumentFragment cho multiple DOM insertions
- ☐ Tránh force synchronous layout (layout thrashing)
- ☐ Implement timeout cho các operations có thể chậm

TỐI ƯU THIRD-PARTY SCRIPTS

- ☐ Audit tất cả third-party scripts hiện tại
- ☐ Loại bỏ các scripts không cần thiết hoặc ít sử dụng
- ☐ Async hoặc defer loading cho non-critical scripts
- ☐ Sử dụng resource hints (preload, prefetch, preconnect)
- ☐ Self-host critical third-party resources khi có thể
- ☐ Implement fallbacks cho third-party scripts fails
- ☐ Monitor performance impact của từng third-party script
- ☐ Sử dụng tag manager để kiểm soát script loading
- ☐ Delay loading cho non-essential widgets/tools
- ☐ Negotiate với vendors về optimized script versions

CẢI THIỆN SERVER RESPONSE TIME

- ☐ Tối ưu Time to First Byte (TTFB) dưới 200ms
- ☐ Sử dụng CDN cho static assets
- ☐ Implement server-side caching hiệu quả
- ☐ Tối ưu database queries và kết nối
- ☐ Sử dụng HTTP/2 hoặc HTTP/3
- ☐ Enable GZIP/Brotli compression
- ☐ Tối ưu server configurations (Apache/Nginx)
- ☐ Implement Keep-Alive connections
- ☐ Sử dụng edge computing khi phù hợp
- ☐ Monitor server response times liên tục

TỐI ƯU CSS VÀ RENDERING

- ☐ Minimize render-blocking CSS
- ☐ Inline critical CSS cho above-the-fold content
- ☐ Lazy load non-critical CSS
- ☐ Tối ưu CSS selectors complexity
- ☐ Tránh CSS animations block main thread
- ☐ Sử dụng transform và opacity cho animations
- ☐ Minimize DOM depth và complexity

- ☐ Avoid large, complex layouts
- ☐ Use CSS containment khi có thể
- ☐ Optimize font loading strategy

FRAMEWORK VÀ LIBRARY OPTIMIZATION

- ☐ Audit và cập nhật các frameworks/libraries cũ
- ☐ Tree-shake unused code từ libraries
- ☐ Sử dụng lighter alternatives cho heavy frameworks
- ☐ Implement proper component lazy loading
- ☐ Optimize bundle sizes với webpack/rollup
- ☐ Use production builds (không dùng development mode)
- ☐ Implement proper error boundaries
- ☐ Optimize re-rendering patterns
- ☐ Use memoization cho expensive computations
- ☐ Implement virtual scrolling cho long lists

MOBILE VÀ TOUCH OPTIMIZATION

- ☐ Tối ưu đặc biệt cho mobile devices (CPU yếu hơn)
- ☐ Test FID trên các thiết bị mobile thực tế
- ☐ Optimize touch event handling
- ☐ Reduce JavaScript execution trên mobile
- ☐ Use lighter polyfills cho mobile browsers
- ☐ Implement responsive design efficiently
- ☐ Optimize images cho mobile (WebP, sizes)
- ☐ Test trên connection speeds khác nhau
- ☐ Implement service worker cho offline support
- ☐ Optimize battery usage considerations

MONITORING VÀ CONTINUOUS IMPROVEMENT

- ☐ Thiết lập continuous monitoring cho FID
- ☐ Tạo dashboard theo dõi Core Web Vitals
- ☐ Set up performance budgets
- ☐ Implement automated performance testing
- ☐ Monitor FID theo user segments (mobile/desktop, locations)
- ☐ Track correlation giữa FID và business metrics
- ☐ Set up alerts cho performance regressions
- ☐ Regular performance audits (hàng tháng)

- ☐ A/B test các optimization strategies
- ☐ Document và share performance learnings

TOOLS VÀ TESTING

- ☐ Google PageSpeed Insights regular checks
 - ☐ Lighthouse CI integration
 - ☐ WebPageTest advanced testing
 - ☐ Chrome DevTools Performance profiling
 - ☐ Real User Monitoring (RUM) implementation
 - ☐ Synthetic monitoring setup
 - ☐ Performance regression testing
 - ☐ Cross-browser FID testing
 - ☐ Load testing under different conditions
 - ☐ User experience correlation analysis
-

VỀ CHUYÊN GIA HỒNG MINH

CHUYÊN GIA HỒNG MINH

CEO LIGHT

Hơn 12 năm kinh nghiệm trong ngành Marketing Online, lập trình, SEO, thiết kế đồ họa, chạy quảng cáo, vv...

- Training chuyên sâu về SEO, Google Ads, Quảng Cáo cho hơn 3000+ doanh nghiệp
 - 20+ Khóa tư vấn đào tạo cho doanh nghiệp về Marketing Online
 - Facebook: <https://www.facebook.com/hm.phm>
 - Chi tiết: <https://light.com.vn/minh-hm>
 - Youtube: <https://www.youtube.com/@minhbmchannel2340>
-

Light - Nhanh - Chuẩn - Đẹp