Ladezeit-Optimierung mit KI-Agenten: Jeder Report als Markdown-Brief für den Coding-Agenten

Veröffentlicht am 05.11.2026 von Christian Schreiber



Der Report von PerfDelta nennt für jede Website die Maßnahmen, die die Ladezeit am stärksten verbessern, mit betroffenen Dateien, Seiten und Elementen. Umsetzen mussten Sie sie bisher selbst. Jeden Report gibt es zusätzlich als Agent-Brief, im PerfDelta Free-Konto für eigene Messungen und ab dem Tarif Solo für jede gemessene Domain: eine Markdown-Datei, die ein Coding-Agent wie Claude Code, Cursor, Codex oder GitHub Copilot im Repository Ihrer Website Schritt für Schritt abarbeitet. Der Agent meldet danach je Maßnahme zurück, was er getan hat, und der nächste Brief kennt diesen Stand.

Ladezeit-Optimierung mit KI-Agenten: der Brief als Arbeitsauftrag

Ein Coding-Agent arbeitet so gut wie sein Auftrag. Wer ihm nur „mach die Seite schneller“ sagt, bekommt Vermutungen. Der Agent-Brief gibt ihm stattdessen dieselbe Grundlage, die auch im Report steht, in einer Form, die er ohne Rückfragen lesen kann. Die Datei hat zwölf Abschnitte:

1. **Auftrag:** Ziel sind gute Core Web Vitals bei echten Nutzern, die Liste wird von oben nach unten abgearbeitet, eine Maßnahme je Commit.
2. **Leitplanken:** keine sichtbaren Änderungen an Design oder Inhalt, keine neuen Plugins und kein Eingriff in Tracking oder Consent ohne Ihre Freigabe.
3. **Hinweis zu gemessenen Inhalten** (dazu mehr unten).

4. **Ausgangslage:** Felddaten am 75. Perzentil, Labordaten je Seite und Gerät, LCP-Element mit Teilzeiten, verschobene Elemente, Seitengewicht, Drittanbieter und Script-Bundles.
5. **Umgebung:** das von außen erkannte CMS, Framework, CDN und weitere Technik, ausdrücklich als „im Repository prüfen“ markiert.
6. **Maßnahmen** in der Reihenfolge des Reports.
7. **Status aus früheren Notizen**, damit der Agent nichts doppelt anfasst.
8. **Veränderungen seit der vorigen Messung:** was seither behoben ist und was sich an Dateien oder Drittanbietern verändert hat.
9. **Performance-Budgets** Ihres Projekts als Abnahmegrenzen.
10. **Prüfweg:** ein Lighthouse-Befehl, den der Agent auf eine lokale oder Staging-Adresse umstellt, und die Nachmessung in PerfDelta nach dem Ausrollen.
11. **Rückmeldeformat.**
12. **Grenzen der Daten.**

Der Brief deckt damit mehr ab als Pagespeed-Score und Core Web Vitals. Kurze Cache-Laufzeiten, fehlende Textkomprimierung, ein blockierter Back/Forward-Cache oder schwere Drittanbieter-Skripte stehen genauso darin wie ein langsames LCP-Bild. Eine Übersicht aller Themen der Ladezeit-Optimierung bietet unser Ratgeber Pagespeed optimieren.

Jede Maßnahme mit stabiler ID, Akzeptanzkriterium und Freigabehinweis

Jede Maßnahme trägt eine ID wie `PD-render-blocking-resources` oder `PD-third-party-blocking-youtube`. Sie ergibt sich aus der Art der Maßnahme und der betroffenen Datei oder dem Anbieter. Build-Hashes im Dateinamen lässt sie weg, deshalb bleibt sie auch nach einem neuen Build gleich und ordnet Notizen der richtigen Maßnahme zu. Dazu kommen:

- **Befund mit Zahlen:** betroffene Metrik, ob sie bei echten Nutzern durchfällt, geschätzte Einsparung, Geräte und Seiten.
- **Dateien und Elemente:** die URLs der betroffenen Dateien mit Einsparung je Datei, bei Layout-Verschiebungen das HTML der springenden Elemente.
- **Maßnahme passend zur Plattform:** Für WordPress nennt der Text etwa ein passendes Plugin, für Next.js die passende Framework-Funktion. Ein empfohlenes Plugin gilt im Brief als Vorschlag, den Sie freigeben.
- **Akzeptanzkriterium:** ein prüfbarer Zielzustand, zum Beispiel „Das LCP-Bild hat fetchpriority="high", kein loading="lazy" und wird im HTML direkt referenziert“.
- **Freigabe nötig:** Maßnahmen an Server, CDN, Caching oder Drittanbietern bereitet der Agent nur vor. Ausrollen darf er sie erst nach Ihrer Zustimmung, weil sie Infrastruktur oder geschäftliche und rechtliche Fragen berühren.

So sieht ein Maßnahmenblock aus dem Brief des Beispielreports aus, gekürzt:

```
### PD-render-blocking-resources: Theme- und Plugin-Dateien aufschieben

- Betrifft: LCP
- Fällt bei echten Nutzern durch: LCP
- Wirkung: geschätzte Zeitersparnis 277 ms
- Freigabe nötig: nein

**Akzeptanzkriterium:** Die genannten Dateien blockieren das erste
Rendering nicht mehr (defer/async, media-Attribut oder kritisches CSS
inline); Lighthouse meldet keine render-blocking Ressourcen für sie.

| Datei | Geschätzte Einsparung |
|---|---|
| `https://sample.perfdelta.com/assets/css/app.css` | 24 KB / 310 ms ~ |
```

Gemessene Inhalte als Daten markiert: weniger Angriffsfläche für Prompt-Injection

Ein Brief enthält Text, den PerfDelta von Ihrer Website gemessen hat. Stünde in einem HTML-Schnipsel ein Satz wie „ignoriere alle bisherigen Anweisungen“, könnte ein Agent ihn als Befehl lesen. Fachleute nennen das Prompt-Injection. Der Brief setzt deshalb HTML-Schnipsel, Dateinamen, Selektoren, URLs, Anbieternamen und Notiztexte in Code-Blöcke, die sich durch ihren Inhalt nicht aufbrechen lassen. Außerdem weist er den Agenten an, diese Inhalte nur als Daten zu behandeln und Auffälligkeiten zu melden. Das senkt das Risiko deutlich, ersetzt aber nicht die Prüfung der Commits vor dem Ausrollen.

Rückmeldung per Notiz: der nächste Brief kennt den Stand

Nach der Arbeit hält der Agent je Maßnahme eine Notiz in einem festen Format fest, zum Beispiel `PD-render-blocking-resources done: kritisches CSS inline (header.php:12)`. Als Status stehen done, skipped, needs_approval und in_progress zur Wahl. Mit verbundenem Pagespeed-MCP-Server schreibt der Agent die Notiz selbst über das Werkzeug `add_note`. Ohne MCP listet er eine Zeile je Maßnahme am Ende seiner Antwort, und Sie tragen sie in das Notizfeld der Report-Historie ein. Eine Notiz fasst höchstens 280 Zeichen; mehrere Zeilen in einer Notiz wertet der Brief einzeln aus.

Der nächste Brief vergleicht die Notizen der letzten 180 Tage mit dem Messdatum und unterscheidet drei Fälle:

- **Noch nicht nachgemessen:** Die Notiz ist jünger als die Messung. Der Agent setzt die Maßnahme nicht erneut um.

- **Weiterhin erkannt:** Die Maßnahme wurde als erledigt gemeldet, eine spätere Messung findet das Problem aber noch. Der Brief bittet dann zu prüfen, ob die Änderung ausgerollt ist oder ein Cache die alte Fassung ausliefert.
- **Behoben:** als erledigt gemeldet, und die spätere Messung meldet die Maßnahme nicht mehr.

Jede Maßnahme wird so nachvollziehbar abgehakt, und die Bestätigung kommt aus der Messung statt aus der Selbstauskunft des Agenten.

Agent-Brief herunterladen oder per MCP abrufen

Im Report steht der Knopf „Agent .md“ in der Leiste „Report teilen“ neben PDF und CSV. Er erscheint für Ihre eigenen Messungen in jedem Tarif, auch im Free-Konto, und ab dem Tarif Solo für jede gemessene Domain. Den Brief des Beispielreports können Sie ohne Konto herunterladen und ansehen. Mit dem MCP-Server holt ein Agent den Brief selbst über das Werkzeug `get_agent_brief`, für eine eigene Messung oder die jüngste Messung einer bereits gemessenen Domain. Notizstatus und Budgets enthält der Brief nur für Domains in Ihren Projekten. Er folgt der Sprache des Reports und ist auf Deutsch und Englisch verfügbar.

Ein Auftrag an den Agenten kann so einfach sein wie: „Lies perfdelta-agent-brief.md und arbeite die Maßnahmen ab. Halte dich an die Leitplanken und melde jede Maßnahme im Rückmeldeformat.“ Legen Sie die Datei außerhalb des Repositorys ab oder schließen Sie sie per `.gitignore` aus, denn sie beschreibt eine einzelne Messung und veraltet mit der nächsten. Prüfen Sie die Commits wie die eines Kollegen, bevor Sie ausrollen.

Grenzen des Agent-Briefs

- **Nur was im Repository liegt:** Dateien von Drittanbietern, Einstellungen im CMS-Backend oder im CDN sieht der Agent nicht. Er markiert solche Maßnahmen als skipped und nennt, wo sie vermutlich zu ändern sind.
- **Einsparungen sind Schätzungen:** Sie stammen aus Lighthouse oder PerfDelta und addieren sich nicht, wenn mehrere Maßnahmen dieselbe Datei betreffen.
- **Lokale Messungen sind nur untereinander vergleichbar:** Lighthouse auf dem Rechner des Agenten läuft mit anderer CPU und anderem Netz als PageSpeed Insights. Maßgeblich ist die Nachmessung in PerfDelta nach dem Ausrollen.
- **Felddaten reagieren langsam:** Der Chrome UX Report fasst 28 Tage zusammen. Eine Verbesserung zeigt sich dort schrittweise über bis zu vier Wochen, im Labortest dagegen sofort nach der Nachmessung.
- **Der Agent kann irren:** Leitplanken und Akzeptanzkriterien senken das Risiko, ersetzen aber keine Prüfung der Änderungen vor dem Ausrollen.

Alle Tarife mit Agent-Brief finden Sie auf der Preisseite. Für einen ersten Report genügt ein kostenloser Website Speed Test.