

Web Performance: Wieso und wie?

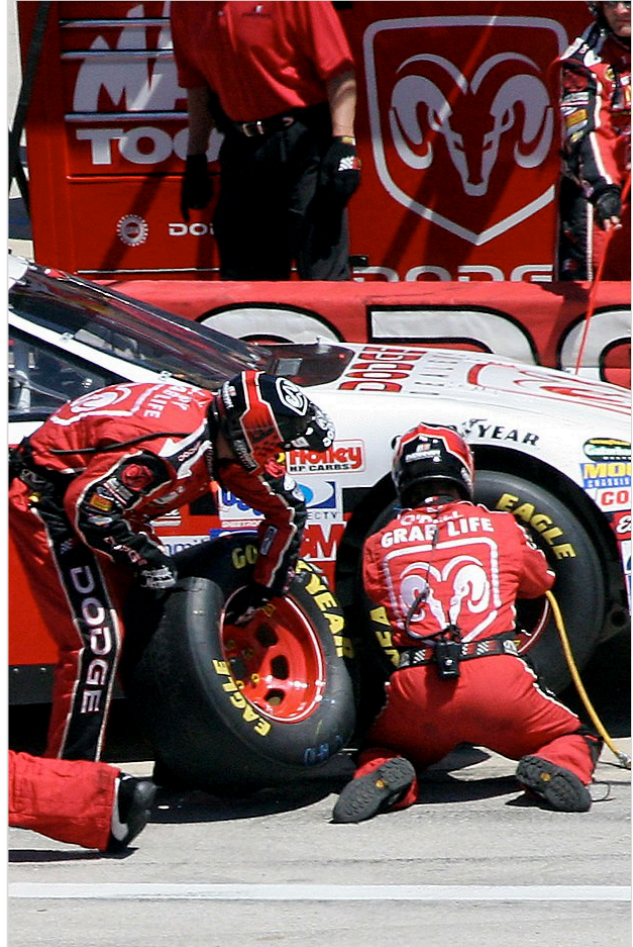
Agenda

Wieso?

Metriken

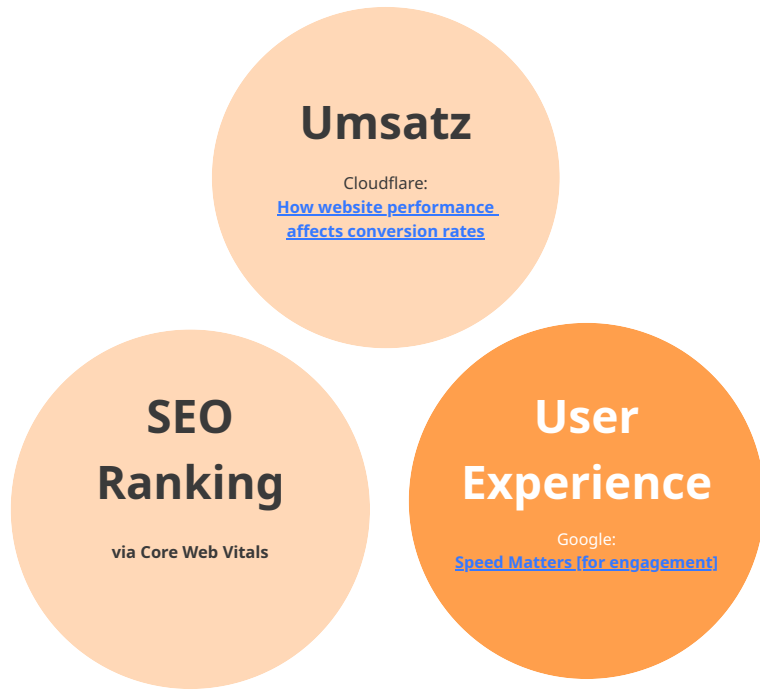
Tools

Praxis



**Wieso
Web Performance
wichtig ist**

Web Performance beeinflusst...



Web Performance Metriken

0s

URL in der Browserleiste eingeben und Enter oder: auf einen Link geklickt, ein Formular abgeschickt...

Verbindungsaufbau:

DNS Lookup
Verbindungsaufbau
TLS Negotiation

Server schickt die Response zurück:

Datenbankzugriffe
Logik
HTML generieren
...

TTFB

Time to First Byte: Erstes Byte der Antwort vom Server kommt beim Client an

TTFB

Time to First Byte: Erstes Byte der Antwort vom Server kommt beim Client an

Laden render-blockierender Ressourcen:
CSS, JS...

Laden von Fonts

FCP

First Contentful Paint: Erste Teile des angeforderten Inhalts werden angezeigt

FCP

First Contentful Paint: Erste Teile des angeforderten Inhalts werden angezeigt

**Laden & Anzeigen vom
Largest Content Element**

Oft ein Hero-Bild oder der Cookie Banner

LCP

Largest Contentful Paint: Größtes Element im Viewport wurde gerendert



Core Web Vital:
Beeinflusst das Google Ranking

Ziel:
< 2.5s

☆ **LCP**

Largest Contentful Paint: Größtes Element im Viewport wurde gerendert

JavaScript
Hydration
Event Loop Blocking

TTI

Time to Interactive: Die Zeit bis die Seite User Input entgegennimmt



Core Web Vitals:

Beeinflussen das Google Ranking



LCP

Largest Contentful Paint:

Größtes Element im Viewport wurde gerendert



CLS

Cumulative Layout Shift:

Layoutverschiebungen beim weiteren Laden



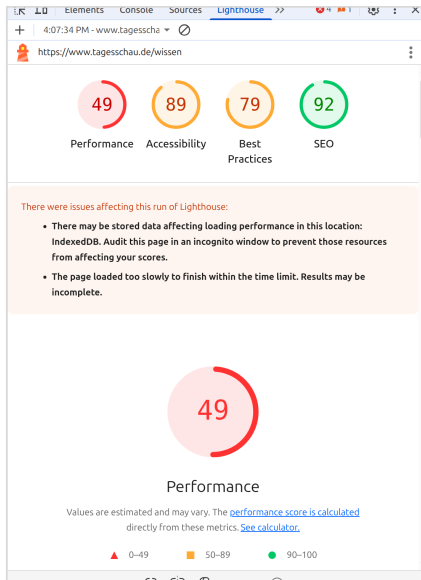
INP

Interaction to next Paint:

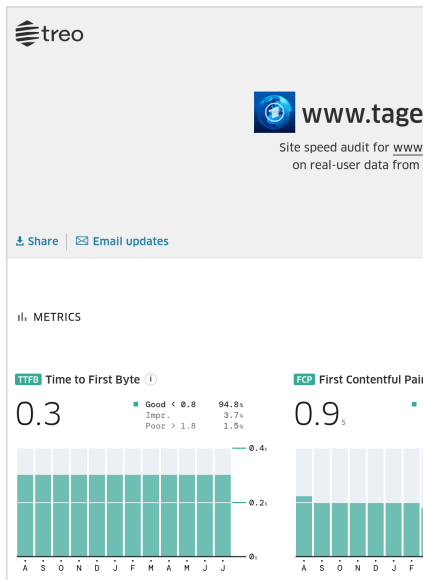
Die Zeit bis die Seite visuell auf User Interaktionen reagiert

Tools

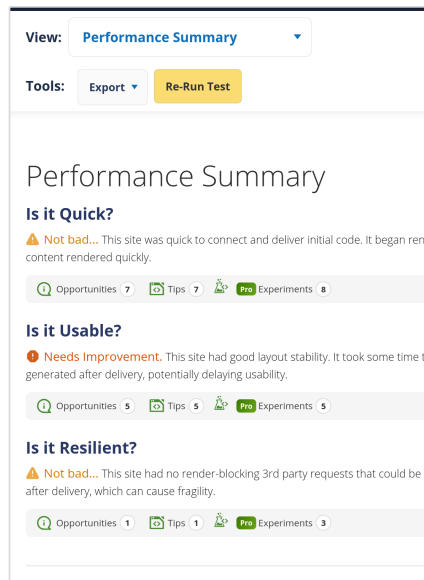
Tools



Chromium-basierte Browser



treo.sh



webpagetest.org

Maßnahmen

Wir setzen uns Höchstwerte,
um Regressionen zu vermeiden:

- **Zeitbasiert:** z.B. auf Basis einer der Metriken
- **Mengenbasiert:** z.B. File Transfer Size (JS, Bilder...)
- **Regelbasiert:** z.B. Lighthouse Score

Auf diesen Werten definieren wir Alerting,
oder lassen den Build fehlschlagen.

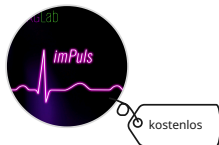
In der Praxis

Bei komoot

Wir konnten bei der komplexen React Anwendung:

- **LCP** von über 3 Sekunden auf 2,6 Sekunden verringern
- **INP** konstant auf unter 200ms bringen
- **Build** schlug fehl, wenn das JS-Bundle zu groß wurde

Noch mehr zu Web Performance?



imPuls

Anhand eines praktischen Beispiels werfen wir gemeinsam einen Blick auf Performanceanalysen und -optimierungen. Anschließend nehmen wir uns Zeit für mitgebrachten Fragen und Beispiele.



20% Rabatt mit
PERFORMANCE_25

Workshop



Quick Check

Mit unserem kostenlosen Quick Check analysieren wir die Performance eurer Website und identifizieren konkrete Verbesserungsmöglichkeiten.

socreatory.com/de/trainings/web-performance

swaglab.rocks/web-performance

Was kann ich jetzt machen?

Schauen wir noch mal
auf den Ladevorgang

0s

URL in der Browserleiste eingegeben und Enter oder: auf einen Link geklickt, ein Formular abgeschickt...

Verbindungsaufbau:
DNS Lookup
Verbindungsaufbau
TLS Negotiation

HTTP
2/3

Latenz
verringern

Server schickt die Response zurück:
Datenbankzugriffe
Logik
HTML generieren
...

Caching

TTFB

Time to First Byte: Erstes Byte der Antwort vom Server kommt beim Client an

Laden render-blockierender Ressourcen:
CSS, JS...

Compression

Laden von Fonts

img
width &
height

FCP

First Contentful Paint: Erste Teile des angeforderten Inhalts werden angezeigt

**Laden & Anzeigen vom
Largest Content Element**
Oft ein Hero Bild oder der Cookie Banner

Preloading

☆ LCP

Ziel:
< 2.5s

Largest Contentful Paint: Größtes Element im Viewport wurde gerendert

JavaScript
Hydration
Event Loop Blocking

TTI

Time to Interactive: Die Zeit bis die Seite User Input entgegennimmt

**Core Web Vitals:**

Beeinflussen das Google Ranking

☆ LCP

Largest Contentful Paint:

Größtes Element im Viewport wurde gerendert

☆ CLS

Cumulative Layout Shift:

Layoutverschiebungen beim weiteren Laden

☆ INP

Interaction to next Paint:

Die Zeit bis die Seite visuell auf User Interaktionen reagiert

Weniger
JavaScript