

Website Performance 2021 – Warum langsame Webseiten euch mehr kosten als schnelle.

Einleitung

Eine schnelle Webseite ist heutzutage Pflicht. Nutzer erwarten Ladezeiten von unter 3 Sekunden. Dabei sind diese vermehrt auf mobilen Endgeräten unterwegs, was viel Frust für Webseitenbetreiber zur Folge haben kann. Denn eine mobile Datenverbindung ist meist deutlich langsamer als DSL und da wird es manchmal schwer ein positives Nutzererlebnis zu erzeugen.

Dazu kommt, dass Google mit den neuen Rankingfaktoren (Core Web Vitals) und dem Mobile-only-Index den Webseiten Betreibern die Pistole auf die Brust setzt und langsame Webseiten abstrafft.

Für die gleiche Konversion musst du deutlich mehr ausgeben, wenn du eine langsame Webseite betreibst.

Benutzt du die richtigen Tools zum Testen?

Meine 2 Favoriten:

- Chrome DevTools
- web.dev/measure

Weitere nützliche Tools um die Performance zu berechnen

- webpagetest.org
- gtmetrix.com
- tools.pingdom.com

Hast du das Performance Budget berechnet?

Du möchtest, dass deine Seite in unter 3 Sekunden für Mobile 3G lädt? Dann darf deine Seite nicht größer als 290kb groß sein.

Dies kannst du ganz einfach unter <https://www.performancebudget.io> berechnen.

Wird dein Inhalt statisch gespeichert und ausgeliefert?

Normal wird eine Seite zur Laufzeit generiert, was einiges an Ressourcen kosten kann. Wenn der HTML Output jedoch in eine statische Datei geschrieben wird und dieser stattdessen jedes mal ausgeliefert wird, werden alle nachfolgenden Anfragen früher durchgeführt.

Beispiel:

Dynamisch generiertes HTML wird in 1,5 Sekunde gerendert und ausgeliefert. Erst im Anschluss können JS, CSS und Bilder geladen werden.

Statischer Inhalt wird in 100ms ausgeliefert, somit laden alle weiteren Anfragen 1,4 Sekunden früher.

Verwendet deine Seite HTTP/2?

Mit HTTP/2 können mehrere Dateien gleichzeitig über eine Verbindung übertragen werden, womit du etwa 20% Zeit einsparen kannst.

Website Performance 2021 – Warum langsame Webseiten euch mehr kosten als schnelle.

Deine Seite hat wenig HTTP-Requests?

Auch mit HTTP/2 solltest du versuchen die Anzahl der Request so gering wie möglich zu halten und nur das zu laden, was für den ersten ViewPort notwendig ist.

Ist dein JavaScript / CSS zusammengefügt und komprimiert?

JS und CSS Dateien sollte zu je einer Datei zusammengefügt und komprimiert werden. Dies spart bis zu 50% Dateigröße. Bei größeren Projekten kann es sinnvoll sein verschiedene kleinere Bundles zu definieren, um ungenutzten Code zu vermeiden. CSS kleiner als 30kb kann inline in den HTML Code eingebunden werden, um sofort geladen zu sein.

Werden Bilder und andere Ressourcen nachgeladen?

Bilder sollten mittels lazyload erst geladen werden, wenn sie in den sichtbaren Bereich gelangen. Dadurch spart man häufig 80% der Datenmenge. Gleiches kann für andere Ressourcen wie JS, CSS oder Fonts sinnvoll sein.

Sind deine Bilder für Web optimiert?

Ein JPG mit 600x300 und 300kb kann z.B. verlustfrei auf 60kb verkleinert werden. Zusätzlich kannst du deine Bilder in WebP ausliefern, was weitere 20kb einspart. Bilder für Web speicher: <https://tinypng.com>

Werden deine Bilder responsive ausgegeben?

Ein mobiles Endgerät hat eine andere Größe als ein 24" Monitor. Daher ist es notwendig verschiedene Bildgrößen für verschiedene Bildschirmauflösungen zu definieren. Dies kann über das <picture> Tag geschehen. Dadurch bekommt der Nutzer immer den perfekten Bildausschnitt und du sparst wieder Datenmenge.

Hast du die Cache Header korrekt gesetzt?

Wenn die Cache Header korrekt gesetzt sind, muss der Nutzer beim zweiten Aufruf deiner Seite nicht lange warten, denn die Ressourcen werden aus dem Browsercache geladen.

Werden deine Dateien komprimiert?

Wenn deine Dateien komprimiert werden (z.B. gzip) kannst du mehr als 60% Datenmenge einsparen. Dabei werden HTML, JS und CSS Dateien komprimiert an den Browser übertragen und vom Client entpackt.

Werden kleine Bilder als base64 ausgespielt?

Man kann kleinere Bilder (< 10kb) mittels base64 inline in den HTML Code einbinden. Dadurch spart man zusätzliche Anfragen an den Server. Kann für kleine Icons oder das Logo sinnvoll sein, damit diese sofort sichtbar sind.

Setzte du aktuelle Software Versionen ein?

Beispiel: stellt man WordPress 5.6 von PHP 7.1 auf PHP 7.4 erhält man einen boost von etwa 60%

Website Performance 2021 – Warum langsame Webseiten euch mehr kosten als schnelle.

Wichtige Inhalte werden über HTTP/2 Server push oder preload geladen?

Wichtige Ressourcen, die für das Rendering benötigt werden, sollten so früh wie möglich geladen werden. Dies kann über HTTP/2 Server push oder preload gesteuert werden.

Dein Webhosting passt zu deinem Projekt?

Nur mit dem passenden Webhosting Paket kann eine schnelle Webseite gewährleistet werden.

Du verwendest bereits einen Proxy Cache oder CDN?

Für größere oder internationale Projekte ist ein CDN oder Proxy Cache sinnvoll um die Last vom Backend-Server zu nehmen und dem Benutzer über einen naheliegenden Server zu bedienen.

Beispiel: Webserver steht in Deutschland, Benutzer kommt aus China, beträgt die Ladezeit dadurch 20 Sekunden. Ist über das CDN ein Server in China, beträgt die Ladezeit unter 3 Sekunden.

Verwendest du Service Worker?

Mit Service Workern kannst du gerade für mobile Endgeräte die häufiger deine Seite besuchen die Geschwindigkeit erhöhen, da der Service Worker auch gecachte Inhalte ausliefern kann, wenn die Internetverbindung unterbricht.

Du hast deine Unterseiten getestet?

Vielleicht gibt es ein kritisches Modul nur auf einer Unterseite? Es reicht nicht, nur die Startseite zu analysieren. Idealerweise testest du jeden Seitentypen gesondert.

Du testest deine Webseite regelmäßig?

Mit jedem neuen Feature solltest du deine Seite neu bewerten. Sonst wunderst du dich vielleicht, warum deine Performance nach einem halben Jahr im Keller ist.

Fazit

Manchmal muss man Dinge ausprobieren, bis man den richtigen Weg für sein Projekt gefunden hat aber wenn du diese Checkliste bei deinem nächsten Projekt berücksichtigst, wird deine Webseite mit hoher Wahrscheinlichkeit ein Erfolg. Für den letzten Feinschliff muss jedes Projekt ganz individuell betrachtet und optimiert werden. Performance Optimierung ist ein dauerhaftes Projekt und wird nicht einmalig durchgeführt.

Du hast noch Fragen? Dann melde dich gerne bei mir.

Markus Amalaraj
markus@digitly.de

Leitung Webentwicklung
digit.ly GmbH