morningtrain

6 TING

DU SELV KAN GØRE FOR AT HASTIGHEDSOPTIMERE DIT WEBSITE / WEBSHOP

INDLEDNING

I takt med at folk vænner sig til en bedre og bedre brugeroplevelse på internettet, så betyder hastighed mere og mere for dit sites performance.

Google tilskriver fortsat hurtige website en enorm værdi, og det gør de, fordi de forstår, hvordan dine brugere agerer på et website. Så ikke blot får du bedre rangeringer i Google ved at have et hurtigt website - du hæver også din konverteringsrate.

To store statements, vi med glæde ville lægge hovedet på blokken for.

Du bør optimere på dit sites hastighed i hvert fald en gang hver 5.-6. måned. Der bliver lagt en del filer op på sådan et site, så sitet sløves løbende.

Du kommer nok aldrig uden om at skulle have fat i en udvikler på et tidspunkt, når der skal optimeres. Men du kan starte et sted selv. Vi har gravet lidt og fundet de 6 ting, som, vi empirisk kan se, øger hastigheden, og som en ikke-udvikler bør kunne implementere selv.

Vores tips gælder både websites og webshops, så for nemheds skyld vil ”sites” fremadrettet dække over begge termer.

Tak, fordi du vil læse med, og god læselyst!

INDHOLDSFORTEGNELSE

1

OPGRADER DIT HOSTING-SETUP

En synder vi tit ser, når et site loader langsomt, skyldes serverløsningen. Specielt hvis du har et stort site med mange funktionaliteter, eller hvis du har meget trafik (30.000+ sessioner om måneden).

DELT VS. DEDIKERET SERVER

Først og fremmest skal du sikre dig, at du ikke har en delt server. Er du hostet hos en af de billigere løsninger (under 100 DKK pr. md.), kan en langsom loadtid skyldes, at du er på en delt server.

En delt server betyder, at der typisk ligger flere sites på samme server som din. Med andre ord kan der være hostet flere hundrede eller tusinde websites på den server, hvor dit website også ligger. Ligger dit site på en server, hvor der er et andet og større site med en masse trafik, som trækker al kapacitet på serveren, så vil det sløve dit site.

DELT SERVER

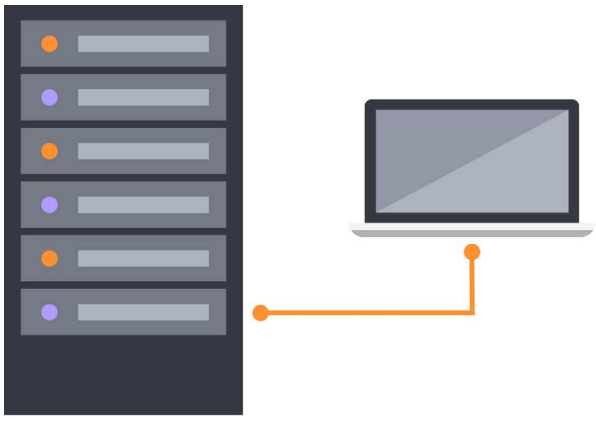
Det kan altså betyde, at du deler server med 5 andre sites, hvoraf 1 af dem kan bruge 50% af kapaciteten. Så må de sidste 4 deles om, hvad der er tilbage. Hvis et af de andre sites har en side, som skal hente en masse kode,

så skal det indlæses fra en server, hvilket også påvirker hastigheden.

Så hvis de andre sites har mange linjer kode, som skal hentes fra serveren, hvad enten det skyldes klodset programmering, eller bare at sitet har mange funktionaliteter, så sløver det dit site.

En hel anden ting er, at hvis en hacker får adgang til et af de andre sites på din server, så kan han få adgang til serveren og dermed også dit site. Så der er ikke blot tale om et hastighedsproblem, men også et sikkerhedsproblem i at ligge på en delt server.

Så kontakt din udbyder, og få opgraderet til en dedikeret server – det betyder, at dit site får sin egen server.



DEDIKERET SERVER

STØRRELSE PÅ SERVER

En server er stort set bare en ekstern computer med et operativsystem (Linux, Windows etc.), hvorpå der ligger en masse filer, som udgør dit site med dets funktionaliteter. Og som du sikkert kender fra hjemmepc'en, så afhænger hastigheden af, hvor stærk en computer du har, og hvor fyldt harddisken er.

Denne analogi kan sagtens overføres på servere, for de har nemlig også CPU'er, RAM og hukommelse.

Hvis du f.eks. har tilføjet for tunge filer på en for lille server, så sløves sitet

ned, da din CPU skal bruge 100% af sin kapacitet. Dette bliver kun værre, alt efter hvor meget trafik du har, da hver bruger på dit site vil tage sin bid af kapaciteten på din server.

Det er, fordi din server kan skrive og læse data med et antal Mbit/s, og hver gang der er en besøgende på sitet, så skal filerne hentes fra serveren til den besøgendes client (dvs. deres computer).

Så har I tunge eller mange filer på serveren, som skal hentes ved besøg på jeres website, og har I meget trafik på sitet, så vil det med garanti sløve sitet umådeligt meget - og i spidsbelastningsperioden sågar gå helt ned.

Endnu en gang ville jeg tage fat i min hostingudbyder for at høre, om sitets størrelse matcher serverens kapacitet. Mange hostingudbydere har et dashboard, hvor du kan se, hvor meget kapacitet dit website bruger på deres server, så start eventuelt med at tage et kig der.

Hvis det ser ud til, at din server er meget belastet, så kan det tyde på, at noget er galt, eller du skal have en større server.

TRAFIKMÆNGDE

De fleste server-setups kan faktisk håndtere en del trafik, hvis bare der er tale om, at serveren skal håndtere lettere filer.

Men hvis man nu bare har et stort site f.eks. en webshop med 10.000 varenumre, så kommer du ikke uden om, at sitet bliver tungere med de ekstra linjer kode og produktbilleder, der skal hentes fra serveren. Dette kommer selvfølgelig an på, hvordan din kode er skrevet, og hvordan du henter data fra din database.

Hvis man så ovenikøbet har mange sessioner på sit site (dvs. besøgende), så betyder det jo, at disse filer skal hentes oftere.

Det er derfor, at store mængder trafik på dit site kan sløve sitet ned og i værste tilfælde få det til at gå ned, fordi serveren skal hente en masse filer enormt mange gange.

Tænk på det som en ligning, der ser sådan ud:

I sådanne tilfælde anbefaler vi, at du hører din host ad, om de kan kigge i serverens historik og se, om den har været overbelastet. Hvis ja, så skal den jo opgraderes – det kommer I ikke uden om. Hvis nej, så er det højst sandsynligvis ikke serverens størrelse, der er problemet.

Hvis du er en sæsonpræget forretning, så anbefaler vi, at du blot i spidsbelastningsperioden opgraderer din serverkapacitet. Det kan f.eks. være Black Friday eller Valentinsdag.

Et server-setup med load balancing ville dog også kunne hjælpe på spidsbelastninger, hvor man så skalerer horisontalt i stedet for vertikalt.

2

MINIMER BILLEDER PÅ SITET

Billeder på sites har altid været den store synder, og det er i virkeligheden et godt sted at starte sin hastighedsoptimering. Som afsnittet ovenfor redegjorde for, så betyder størrelsen på de filer, klienten skal hente fra serveren, noget, samt antallet af gange disse filer skal hentes.

Det er derfor meget afgørende for, hvor hurtigt dit site loader. Vi ser typisk, at de største filer på sites er billeder. Specielt moderne sites er bygget op omkring store billeder i høj opløsning, hvilket kan sløve dit site helt ekstremt.

Men hvad skal man så gøre? Skal man bare nøjes med små grynede billeder?

NEJ, selvfølgelig skal man ikke det. Man kan bare vælge at optimere sine billeder, hvilket man kan gøre uden at miste kvaliteten af billederne.

FYSISK STØRRELSE PÅ BILLEDERNE

F.eks. ser vi ofte, at der er uploadet billeder, som er 2000 pixel x 1500 pixel, men på sitet bliver de brugt i en størrelse på 650 pixel x 380 pixel.

Det, der helt lavt praktisk sker her, er, at sitet henter den store fil på 2000 pixel x 1500 pixel fra serveren, skalerer billedet ned til størrelse 650 pixel x 380 pixel, så billedet passer ind på sitet, og indsætter den der, hvor den hører til.

Det betyder altså, at filen, der reelt set hentes fra serveren, er 3 gange større end den, slutbrugerne i sidste ende ser. Den faktiske størrelse (antallet af pixels) på billedet hænger direkte sammen med MB-størrelse på filen, som skal hentes.

Man kan altså derfor opnå store hastighedsforbedringer ved at minimere sine billeder til en mere passende størrelse. Det kan man gøre i InDesign, Photoshop eller ved at bruge et online tool.

En Google-søgning på ”compress image” giver et hav af resultater, som både indeholder tools og plugins, der kan løse problemet.

BILLEDETS STØRRELSE
PÅ DIT WEBSITE

BILLEDETS STØRRELSE
I VIRKELIGHEDEN

KVALITETEN AF BILLEDET

Jo større billedet skal vises, jo højere opløsning bør det være i. Det betyder altså, at de billeder, som man mindsker i størrelsen, godt kan tåle at blive neddroslet i opløsning, uden at dine slutbrugere vil kunne se en forskel.

Ved at smide dine billeder igennem et tool som f.eks. tinypng.com, så kan du optimere sine filstørrelser med helt op til 60%, uden at miste kvaliteten på billedet.

Det blotte øje vil ikke kunne se forskel på, om der er brugt 1000 farvevarianter eller 500. Antallet af farvevarianter, der bruges, gør dog en stor forskel for størrelsen af filen.

Vi ser tit problemet på webshops, hvor der er hundredevis af produktbilleder. Disse vises for brugeren i meget små formater, og derfor bør man ikke uploade store produktbilleder til serveren.

Med x antal varenumre får man hurtigt hæmmet sit sites hastighed, f.eks. hvis man har en produktkategoriside, hvor der loades 10+ produktbilleder. Dette leder os til punkt 3, nemlig lazy-loading.

#3

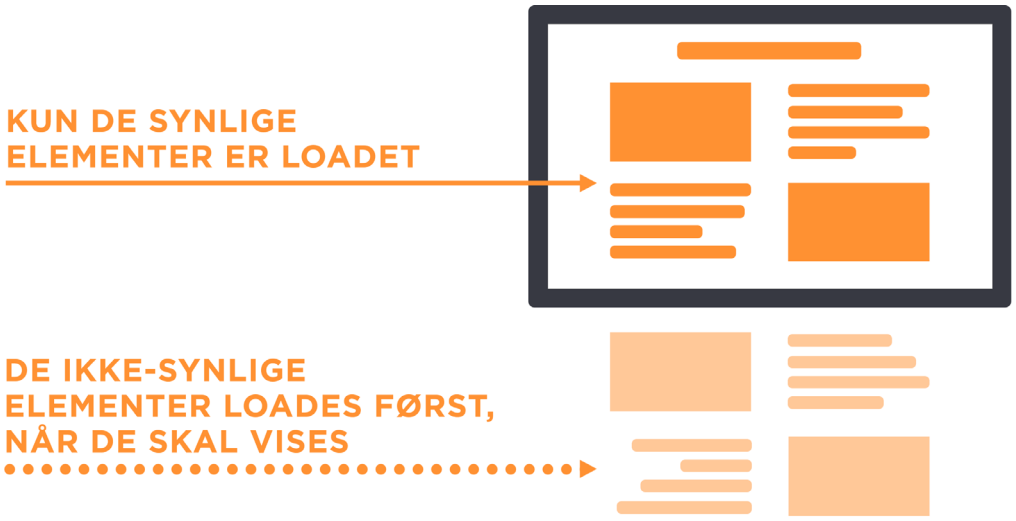
UDSKYD INDLÆSNING AF IKKE-SYNLIGE ELEMENTER

Ja, med denne kommer du nok ikke uden om en udvikler. Så har du ikke adgang til sådan en, så kan du springe den over. **BEMÆRK DOG**: Dette indgreb er oftest det mest effektive efter optimering af billeder.

Et billede begynder at tegne sig: Sitet bliver sløvet af serverens kapacitet, hvis der er mange større filer, der skal hentes ind, og hvis det skal hentes ofte grundet høj trafik. Derfor kommer vi til dette tredje trin – udskyd indlæsning af elementer.

For når man først har forstået, hvor meget filhentningen betyder, så kan man med fordel arbejde med, at hele sitet loades ikke ind, når man forsøger at tilgå det.

Man kan (relativt) nemt programmeringsmæssigt lave et indgreb, der gør, at det kun er de elementer, der er synlige for brugeren, der bliver er loadet. I takt med at brugeren navigerer rundt på sitet eller scroller, så indhentes filerne, der skal vises for dem – dette er bedre kendt som lazy loading, også kendt som on-demand loading.



Ved at udskyde indlæsningen af elementer fra serveren, indtil de skal bruges, så er det absolut kun de filer, som rent faktisk skal bruges, som hentes.

På den måde sparer man en hel masse serverkald, hvor ”unødvendige” filer ikke hentes. Og som de to forrige afsnit har lært os, så betyder det jo kort sagt, at vi nu har sørget for, at vores store filer skal hentes sjældnere = et hurtigere site og en bedre brugeroplevelse.

NOTE: Man kan blive fristet til bare at gå amok og sørge for, at alt indlæses, i takt med at folk scroller. Her vil vi på det kraftigste anbefale, at man tænker over, hvilke elementer er tunge at hente, så man kun tilføjer lazy load på de ressourcer/filer, som fylder meget. Mange større virksomheder har implementeret en skeleton screen.

Tjek f.eks. youtube.com - her får du vist skelettet først, inden videoer loades, da Google giver dette en vis værdi.

ADVANCED TIP

Man skal i princippet gøre, hvad man kan for udskyde indlæsning af render-blocking stylesheets og tredjepart scripts. Vi ser tit, at indlæsning af mange tredjepartsscripts sløver dit site, så sørg for, at du kun henter tredjepartsscripts og eksterne stylesheets på de sider, hvor du reelt skal bruge det.

Hvis man er interesseret i at læse mere om First Contentful Paint og First Meaningful Paint, så er Google en god kilde.

4

OPSÆT CACHING

Nu har vi forstået, at det hele i virkeligheden handler om:

- Hvilke filer, der skal hentes
- Hvor ofte de skal hentes
- Om det er unødvendige filer, der hentes

Dertil kommer næste løsning: Opsætning af caching. Caching er en funktionalitet, hvor data gemmes, så indhold hurtigere kan tilgås næste gang, det besøges. Et eksempel kunne være caching på en klients enhed.

Her ville man kunne gemme data i browsercachen. Din webbrowser cacher typisk HTML, JavaScript og billeder. Dette gøres for, at du som klient hurtigt kan få tilgang til data og filer på hjemmesider, som du har besøgt før.

Ideen med caching er altså, at man gemmer data, som er hentet fra serveren ned på klientens (dvs. brugerens) computer, og derved skal færre data hentes fra serveren, når personen atter besøger jeres site.

Det hele lyder mere teknisk end som så, men sandheden er, at der findes gode plugins, som kan klare det for dig i f.eks. WordPress og Magento.

Du skal derfor ikke have den helt store udviklerkasket på for at lykkes med caching på website-niveau, hvilket nok er den mest hyppige form for caching. Udfordringen kan dog være, hvis du ønsker at konfigurere, hvilke data der skal caches og hvordan. I så fald kræver det, at du har noget viden om din data.

Caching findes dog i flere lag/niveauer. De er kort ridset op nedenfor.

CLIENT-SIDE CACHE

Ved client-side caching kigger man typisk på caching i browseren og i HTTP Cache headers. Denne form for caching foregår på klientens enhed. Det gør det muligt, at du kan gemme en session på en bruger.

Et eksempel kunne være, at browseren gemmer data på en kurv du har fyldt

på en webshop. Du lukker derefter webshoppen, og genbesøger den. Når du genbesøger webshoppen, så vil dine varer stadig ligge i kurven.

DNS

DNS-cache står for at cache DNS-rekorder. Dette er med til at øge hastigheden, når du skal hente information.

Du kan tænke på dette som en slags telefonbog. Du behøver ikke huske telefonnumre, da du bare slår op i din telefonbog.

DNS-cachen fungerer på samme måde; den gemmer IP-adresser på websites, så næste gang jeg kigger efter morningtrain.dk, så ved min DNS-cache, at IP-adressen er 104.26.15.14, og den slipper for at slå DNS op.

Der foregår også caching på database- og applikationsniveau. Her foregår caching typisk bare ved hjælp af key-value data stores. Konceptet er i og for sig det samme. Det handler om at gemme data, så man skal requeste data fra den originale kilde hver gang, man skal bruge det.

FORDELE VED CACHING

Et par fordele ved at cache er bl.a., at man øger en sides performance markant, da de lokale RAM på en computer opererer væsentligt hurtigere end disken på en server.

Derudover så kan det sænke ens serverbelastning markant, da man ved at gemme data lokalt på den besøgendes client sparer en masse serverkald, så serveren ikke skal ”arbejde” for at levere data = mindre CPU-forbrug.

Det betyder ganske enkelt en bedre brugeroplevelse, og en bedre brugeroplevelse øger din konverteringsrate markant. Bare se på Amazon - de mister millioner af dollars, for hvert sekund deres site loader langsommere.

ULEMPER VED CACHING

Der er faktisk kun én enkelt udfordring med caching, og det er, at den foregår lokalt. Det betyder, at hvis du har et site, som folk i høj grad genbesøger, og

du foretager en del optimeringer på dit site, så risikerer du, at de besøgende ikke har den nye version af dit website, så de ser den gamle version af sitet. Det kan f.eks. være, at der var en fejl på dit site, som nu er blevet udbedret. Så skal hver client (brugers computer) have slettet deres browsercache lokalt, før ændringerne kan ses.

Din browser rydder typisk cachen hver 30. dag, så det er altså kun et problem, hvis man laver en del optimeringer/rettelser på sit site, og hvis ens besøgende ofte kommer igen.

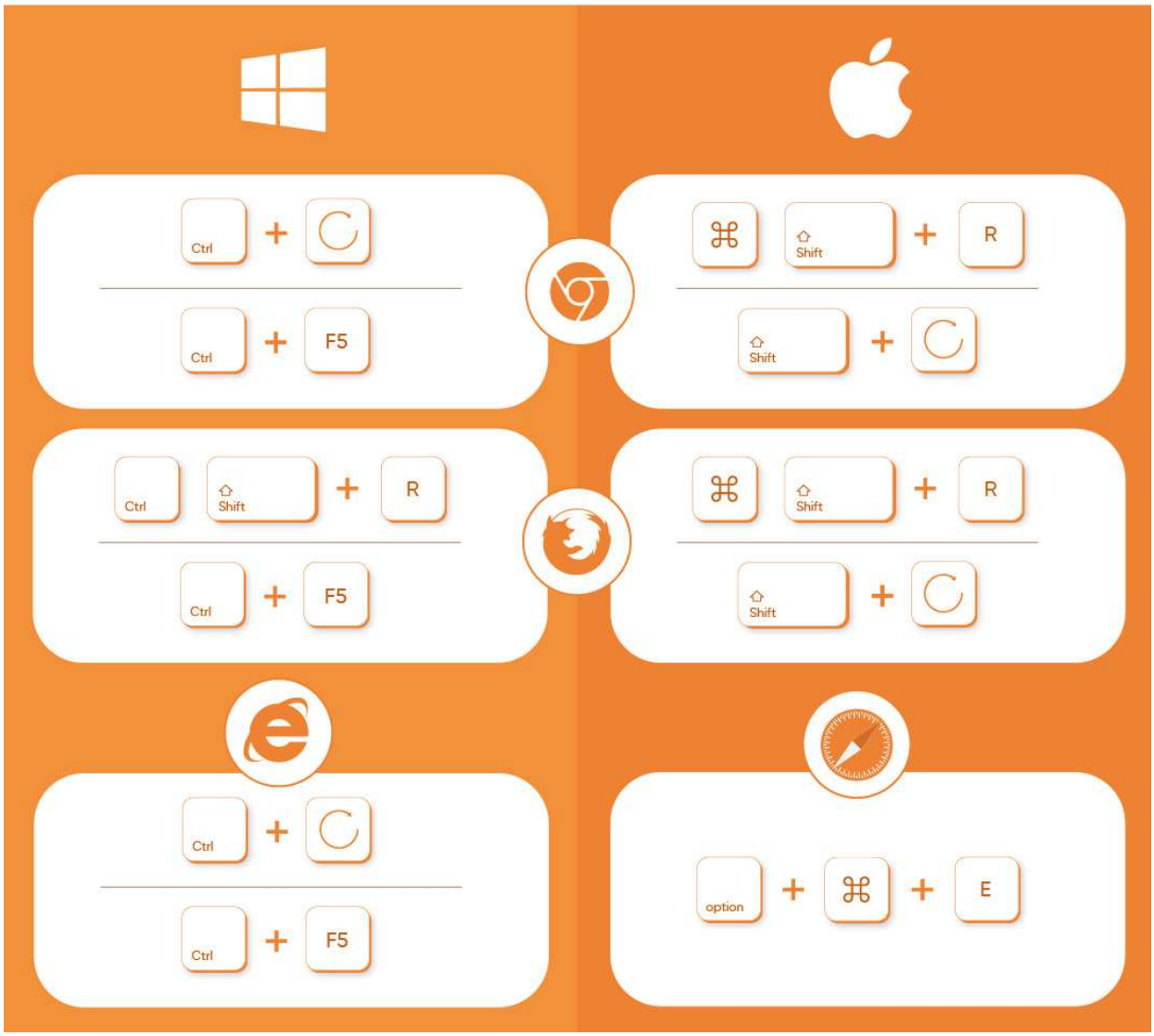
Det er dog muligt ved hjælp af tags at fortælle en browser, hvor lang levetid der skal være på de ting, dit website cacher i en browser.

På næste side finder du en guide til, hvordan du rydder din cache.

ADVANCED TIP

Hvis du ønsker at styre, hvor længe de forskellige ting caches i din browser fra dit website, så skal du kigge på Cache-Control, som skal sættes i HTTP header. På den måde har du mulighed for at styre levetiden på det, der skal caches og eventuelt tvinge, at dit website caches på ny i klientens browser.

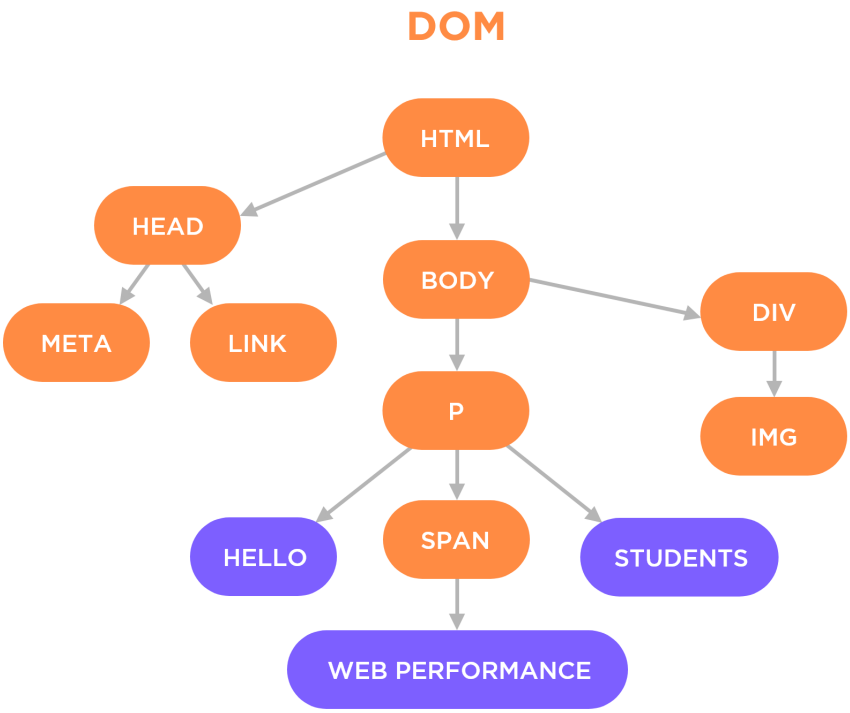
SÅDAN CLEARER DU DIN CACHE



5

UDSKYD LOAD AF IRRELEVANTE SCRIPTS

Vi ser ofte, at sites bliver sløvet ned, fordi der indlæses utallige scripts ind på sitet. Oftest indlæses de alle i <head></head>. Head indeholder metadata omkring dit website. Head-elementet bliver renderet/hentet samtidig med dit <body></body> element, hvilket kan sløve dit site, hvis du ligger dine scripts heri.



Med scripts henviser vi til JavaScript eller .js-filer, som oftest bruges til at manipulere dit website eller indlæse ting fra tredjeparter. Tredjeparter kunne være:

- Google Analytics - https://www.google-analytics.com/analytics.js
- Facebook Pixel - https://connect.facebook.net/en_US/fbevents.js
- Pop-ups (genereret i tredjepartssoftware)
- Leadforensics
- LinkedIn Insight Tag m.m.

ET PAR ØVELSER TIL DIG

1. Kan du selv finde nogen af ovenstående på dit website? Og bliver de loadet?
2. Ligger de i dit <head></head> element eller i bunden af dit <body></body> element?

At alle disse scripts indlæses før sitet i sig selv, er altså ikke optimalt, hvis du ønsker, at dit website skal loade hurtigt. Hver gang browseren skal hente et script, pauser den nemlig, før den kan hente resten af websitet.

Det betyder i praksis, at alle disse scripts bliver indlæst før det webcontent, som brugeren skal se. Så brugeren oplever bare, at der ikke sker noget, når de prøver at indlæse sitet, men i virkeligheden er den i gang med at indlæse en masse scripts, som ikke er synlige for brugeren. Og det, må man sige, er dårlig brugeroplevelse.

Du bør ikke undervurdere, hvor meget scripts kan fylde på dit site. Vi har set flere eksempler, hvor scripts på sitet fylder mere end f.eks. billederne, som plejer at være den helt store synder.

CONTENT TYPE	PERCENT	SIZE
Script	45.25%	890.2 KB
Image	31.54%	620.5 KB
Font	13.85%	272.5 KB
CSS	6.37%	125.4 KB
HTML	2.75%	54.1 KB
XHR	0.24%	4.6 KB
Total	100.00%	2.0 MB

Det vil helt sikkert få din afvisningsrate (bounce rate) til at stige markant, da folk så tænker, at der er noget galt, og så hopper de bare tilbage, hvor de kom fra. Vores anbefaling til dig som ikke-udvikler er at bruge Google Tag Manager (GTM) til at indlæse alle dine scripts. Hvis du ikke allerede ved, hvordan du bruger GTM, så kan du læse en kort guide her.

Denne e-bog skal nemlig ikke handle om GTM 101's, men snarere om, hvordan du kan bruge GTM til at udskyde ikke-væsentlige scripts, til sideindholdet er indlæst.

Hvis du sørger for at indlæse alle dine tredjepartsscripts via GTM, får du ikke kun et tydeligt overblik, over hvor mange scripts der bliver indlæst, men du har også muligheden for at gøre noget ved hastighedsproblemet – uden en udvikler.

MULIGHED NR. 2

Du kan også vælge at indsætte dine eksterne scripts nederst i dit body-tag. Det gør, at eksterne scripts først bliver hentet, når alt andet i dit body-tag er hentet. Det er med til at give en bedre brugeroplevelse, da visningen af websitet vil være hurtigere.

ADVANCED TIP

Hvis du er sikker på. at de JavaScript-filer, som du vil hente, ikke får din DOM til at breake, så kan du tilføje async til dit script ligesom her: <script src="script.js" async></script>. Gør du dette, fortæller du din browser, at den ikke skal blokere din DOM i at blive bygget, når den afventer scriptet. Det kan hjælpe med at øge din loadtid.

6

REFAKTORER, OG RYD OP I KODEN

Som vi nu har belyst et par gange, så er the name of the game: ”Hvor mange og hvor store filer skal hentes fra serveren?” Dette afsnit handler om delen, der hedder ”hvor mange”.

Rigtig mange sites har funktionaliteter og linjer af kode, som enten er overkomplicerede eller slet ikke bør bruges. Vi ser ofte, at kode-filerne er meget store, eller at der er mange af dem sammenlignet med sitets reelle størrelse og funktionalitet.

Med kode tænker vi hovedsageligt på JS, CSS, HTML og Font-filerne. I eksemplet her udgør koden 23% af de filer, der skal hentes.

CONTENT TYPE	PERCENT	SIZE
Script	45.25%	890.2 KB
Image	31.54%	620.5 KB
Font	13.85%	272.5 KB
CSS	6.37%	125.4 KB
HTML	2.75%	54.1 KB
XHR	0.24%	4.6 KB
Total	100.00%	2.0 MB

FONTE OG FARVER

Hvorfor fylder fonte så meget? Jo, det kan virke skørt, men alle sites har en fil, der definerer, hvilken font sitet bruger samt et alternativ, hvis den besøgendes client ikke har adgang til den valgte font.

Visse custom fonte kræver flere KB, da der i filen skal defineres, hvordan alle

bogstaver ser ud. Dvs. ikke blot i **fed**, *kursiv* og <u>understreget</u>, men også som de forskellige overskrifter osv.

Så når du fravælger de mere simple standardfonte, vil du uundgåeligt få en tungere fontfil. Det gør sig også gældende for Adobe-fonte, som færre clienter har adgang til sammenlignet med Google-fonte.

Google Fonts er en database med 17 trillioner fonte at vælge imellem. Disse fonte udgør standarden og kan indlæses på stort set alle devices. Et godt sted at starte er at sørge for, at man bruger en Google-font.

EXTENSIONS OG PLUGINS

Mange sites, som til dagligt styres af en ikke-udvikler, vil over tid få installeret mange plugins, add-ons eller extensions (navnet afhænger af, hvilket CMS du bruger). Det fleste kender ordet ”plugins”, så det kalder vi det herfra.

Plugins kan være både små og store pakker af kode, som eksterne udviklere har lagt tilgængelige på din CMS’ platform. Det er altså eksterne udviklere, som har kodet noget funktionalitet til det givne CMS, som man så downloader og installerer på sit site.

Det, der helt lav praktisk sker, er, at du copy/paster en pakke med x linjer kode, som du så tilføjer til din kodebase for at opnå en bestemt funktionalitet.

Ofte har et plugin meget funktionalitet for at dække flest mulige use cases. F.eks. vil et plugin til at vise en slider i toppen af dit site ofte have 10 forskellige animationer og en masse funktionaliteter, som du ikke bruger eller behøver. Du skulle bare have en simpel slider, som kan skifte to billeder.

På trods af at du kun har brugt 20-30% af pluginnets funktioner, så har du stadigvæk downloadet kodebasen til alle de funktioner, som du ikke bruger. Det betyder altså, at du har hentet en masse PHP, JavaScript og HTML, som hentes fra serveren og ind på sitet, hver gang siden loades – også selvom du ikke bruger størstedelen af koden til noget som helst.

Med andre ord vil disse plugins sløve dit site unødigt samt skabe flere mulige sikkerhedshuller og potentielt tilføje dårlig kode til din kodebase.

Du bør altså gå dine plugins igennem for at se, om der er nogle, som du ikke

bruger. Og hvis du ikke bruger dem længere, er det ikke nok at deaktivere dem - du skal slette dem helt for at fjerne koden fra din kodebase.

Nu er det altså ikke sådan, at du bare skal gå ind og slette alle plugins. Flere af de store plugins er gode, og dem bør du beholde. Har du mindre plugins, der løser småopgaver såsom redirects eller sliderfremvisning, bør du få erstattet dem med kode, og ja, her skal du have fat i en udvikler.

JS, CSS & HTML

Dette afsnit er kun relevant, hvis du har en udvikler ved hånden. Har man rent faktisk fået kodet sitet, så gælder der her det samme som afsnittet ovenfor om plugins.

Man bør gå ind i koden for at se, om der er redundante filer, eller om man kan samle antallet af .css- og .js-filer, som skal hentes fra din server. Således spare du tid ved, at du ikke opretter en ny request, hver gang en .js- eller .css-fil skal hentes.

Det, vi oftest ser, er, at der er nogle linjer kode, som enten overskriver hinanden eller definerer de samme ting. Der er også linjer, som slet ikke gør noget i frontenden længere, da dens funktionalitet måske er blevet fjernet eller erstattet.

MINIFICATION

Der findes flere plugins til at minify dine filer. Minification er en proces, hvor man går ind og minimerer kode og markup på dit site.

Det gøres for at reducere loadtiden af dine filer, da du typisk kan reducere størrelsen på filerne. Det, som sker, når du går ind og minifier din kode, er, at kommentarer og ekstra mellemrum fjernes, og variable navne gøres mindre. Et typisk eksempel kunne være følgende JavaScript-fil, som før minifying ser således ud:

```
// function returns a promise that succeeds if a 6 is tossed
function tossASix() {
  return new RSVP.Promise(function(fulfill, reject) {
    var number = Math.floor(Math.random() * 6) + 1;
    if (number === 6) {
      fulfill(number);
    } else {
      reject(number);
    }
  });
}
// return random number between 1 and 6
function dieToss() {
    return Math.floor(Math.random() * 6) + 1;
  }
// display toss result and launch another toss
```

```
function logAndTossAgain(toss) {
  console.log("Tossed a " + toss + ", need to try again.");
  return tossASix();
}
function logSuccess(toss) {
  console.log("Yay, managed to toss a " + toss + ".");
}

function logFailure(toss) {
  console.log("Tossed a " + toss + ". Too bad, couldn't roll a six");
}
// use promise paradigm to try three times to toss a 6
tossASix()
  .then(null, logAndTossAgain) //Roll first time
  .then(null, logAndTossAgain) //Roll second time
  .then(logSuccess, logFailure); //Roll third and last time
```

Men hvor den efter at være blevet minified ser således ud:

```
function dieToss(){return Math.floor(6*Math.random())+1}function
tossASix(){return new
RSVP.Promise(function(a,b){var c=Math.floor(6*Math.
random())+1;6===c?a(c):b(c)})}function logAndTossAgain(a){return
console.log("Tossed a "+a+", need to try again."),tossASix()}function
logSuccess(a){console.log("Yay, managed to toss a "+a+".")}function
logFailure(a){console.log("Tossed a "+a+". Too bad, couldn't roll a six")}
tossASix().then(null,logAndTossAgain).then(null,logAndTossAgain).
then(logSuccess,logFailure);
```

Som du kan se, fylder ovenstående eksempel meget mindre, for både i linjer, og mellemrum er fjernet. Det gør filen mindre, og derved kan den hentes hurtigere fra serveren.

Der findes mange plugins til at gøre ovenstående, men hvis det skal konfigureres og gøres optimalt, vil vi anbefale, at du tager fat i en udvikler.

Tak, fordi du læste med, vi håber, vores e-bog har givet dig blod på tanden til at komme i gang, så dit site bare kan flyve derudad i topfart.

God arbejdslyst!

OM FORFATTERNE

NICLAS JOHANSEN
Programmeringschef & partner

Til daglig leder Niclas vores backend-afdeling på 10 mand, hvor han er ansvarlig for at optimere afdelingens processer og udvikle de ansattes faglighed.

KONTAKT
Tlf.nr.: +45 42 90 84 28
E-mail: nj@morningtrain.dk

THOMAS ØSTERKJERHUUS
Marketingchef & partner

Thomas er marketingchef, hvor han til daglig udvikler adfærds- og datadrevne vækstrejser for Morningtrains kunder, samtidig med at han udvikler sit teams kompetencer.

KONTAKT
Tlf.nr.: +45 92 44 93 82
E-mail: to@morningtrain.dk

OM MORNINGTRAIN

Morningtrain er et pulserende webbureau, som efter flere år som unge iværksættere i Odense har etableret sig som et af byens førende programmerings- og marketinghuse.

Vi er webnørder, som elsker at programmere og optimere online forretning. Derfor har vi specialiseret os inden for tre kerneområder: Systemudvikling, webdesign og online markedsføring.

Vores passion er at skabe digitale løsninger, som ikke kun virker her og nu, men som også kan mærkes på bundlinjen i det lange løb.

Vores 30 digitale specialister har altid din ryg, og det har vi fået lavet en lækker lille reklamefilm om. Se den her.

KONTAKT
Tlf.nr.: +45 71 99 55 30
E-mail: mail@morningtrain.dk
Website: www.morningtrain.dk