Bluebird Day.
a Handpicked agency

Performance checklist.

Server-side

☑ HTTP2

HTTP/1 of HTTP/2, waar draait je server op? HTTP/2 kan meer requests tegelijk aan terwijl HTTP1 maar 5 requests tegelijk kan verwerken. Dit komt doordat HTTP/2 de requests samenvoegt (multiplexing) waardoor er meerdere bronnen in 1 request worden geladen. Naast het aantal requests biedt HTTP/2 nog meer voordelen waardoor het een vereiste is voor een moderne, snelle hosting.

Door owner en developer.

☑ HTTPS

Niet zozeer een performance metric als een seo metric. Zorg dat je website een SSL certificaat heeft om de data van je gebruikers te beschermen en als ‘veilig’ te worden bestempeld door zoekmachines.

Door owner en developer.

☑ Cache headers

Zorg dat de caching headers voor veel voorkomende bestanden goed zijn ingesteld om te voorkomen dat statische bestanden telkens opnieuw worden geladen. Denk aan dezelfde stylesheet op alle pagina’s, afbeeldingen of icons die in buttons zijn gebruikt.

Door developer.

☑ Onbereikbare bestanden

Zorg ervoor dat er geen requests worden gemaakt naar onbereikbare bestanden. Omdat de browser meerdere ‘round trips’ uitvoert om te bepalen of een bestand bereikbaar is kost een onbereikbaar bestand kostbare tijd. Vergelijk het met een conversatie:

browser: ‘Ik heb bestand genaamd product.jpg uit deze map nodig’
server: ‘Ik heb voor je in die map gekeken, maar vind geen bestand met die naam’
browser: ‘Zeker weten? Ik zie in de HTML dat het bestand in die map moet zitten.’
server: ‘Nee, ik heb het bestand hier echt niet. Error 404.’

Door developer.

HTML

☑ Minificeer HTML

Door het verwijderen van comments, witruimte en onnodige quotes kun je de load van de HTML van de pagina zelf aanzienlijk verminderen.

Door developer.

☑ Laad CSS voor JavaScript

Google heeft getest wat de ideale volgorde wat betreft het laden van CSS en JS is. Omdat de browser met alle lopende processen stopt om een script te laden. Daarom is het beter om CSS prioriteit te geven boven JavaScript.

Door developer.

☑ Beperk iframes

Elk iframe op een pagina is een volledig nieuwe paginarequest binnen de huidige pagina. Iframes kunnen voorkomen bij embedded video’s, maar bepaalde marketing-gerelateerde scripts kunnen resulteren in iframes op de pagina. Het is goed om je development team en je marketing afdeling te laten afstemmen welke iframes geplaatst gaan worden.

Door marketing en developer.

CSS

☑ Minificeer je CSS

Doordat je onnodige witruimte uit een CSS bestand haalt reduceer je de grootte van het bestand. Daarnaast is het mogelijk om css-selectors te ontdubbelen en stijlregels efficiënter te maken. Een beknopt voorbeeld:

```
<style>
    #myContent { font-family: Arial }
    #myContent { font-size: 90% }
</style>
```

wordt na minificatie:

```
<style>#myContent{font-family:Arial;font-si-
ze:90%}</style>
```

De meeste witruimte is weggehaald en de 2 stijlregels (met een zelfde selector) zijn samengevoegd.

Door developer.

☑ CSS koppelen

Voeg alle losse CSS bestanden samen en minimaliseer daardoor het aantal requests. Dit is een overblijfsel uit het HTTP1 tijdperk. Door het gebruik van HTTP2 is dit wat minder relevant geworden. Bij oudere browsers die sowieso geen of gedeeltelijk HTTP2 requests ondersteunen en dus terugvallen op HTTP1.1 is het nog steeds relevant om CSS samen te voegen, dus ga na of dit voor jouw doelgroep relevant is.

Door marketing en developer.

☑ Ongebruikte CSS

Als je al wat langer een website in de lucht hebt kan het voorkomen dat door veranderingen aan de site, bepaalde CSS niet meer relevant is. De hoeveelheid relevante CSS voor een bepaalde pagina noem je CSS Coverage. Door een analyse te doen in Google Chrome Dev Tools kun je bepalen wat je CSS Coverage is.

Door developer.

☑ Critical Path CSS

Ja, het is goed om je losse bestanden samen te voegen, maar een goede optimalisatie is het definiëren van de losse templates van je website en per template te bekijken welke styling nodig is. Nog beter is het om de styling die nodig is voor de content ‘boven de fold’ inline in de head van de pagina te laden. Daarna kun je het volledige stylesheet laden wanneer de pagina geladen is.

Deze techniek heeft Critical Path CSS. In theorie versnelt dit de laadtijd niet, maar gebruikers hebben we sneller een gestylede interactieve pagina tot hun beschikking. Deze techniek heet Critical Path CSS en is een grote verbetering van de perceived performance.

Door developer.

Javascript

☑ JavaScript minificeren

Door de witruimte van binnen je scripts te verwijderen creëer je kleinere bestanden. Veel JavaScript minifiers kunnen de namen van functies en variabelen verkleinen waardoor je bestanden nog kleiner worden.

Door developer.

☑ JavaScript samenvoegen

Door de geminificeerde JavaScript samen te voegen creëer je minder requests naar bestanden.

Door developer.

☑ JavaScript Bundling

Scripts kunnen worden samengevoegd om het aantal requests te beperken. In plaats van het samenvoegen van alle scripting kan ook voor JavaScript Bundling worden gekozen. Dit is vergelijkbaar met het inladen van CSS voor bepaalde templates; je serveert dan enkel de JavaScript die nodig is voor de huidige pagina, waardoor de laadtijd van de pagina weer afneemt.

Verder kunnen scripts asynchroon worden ingeladen. Dus de scripts worden op de achtergrond ingeladen zonder het renderen te blokkeren. Wanneer het script is geladen wordt deze uitgevoerd.

Door developer.

☑ Kleinere JavaScripts inline

Het kan efficiënter werken om kleinere script blokken inline in de body zetten. Zeker wanneer deze scripts pagina-specifiek zijn.

Door developer.

☑ Update JavaScript libraries

Vaak komen er nieuwe versies van externe JavaScript libraries uit. Één van de bekendste hiervan is jQuery. Helaas worden er ook vaak kwetsbaarheden in dergelijke libraries gevonden. Deze kwetsbaarheden worden gelukkig snel gedicht in nieuwe versies. Daarom is het noodzakelijk om tijdig de externe libraries te updaten. Dit wordt bijvoorbeeld meegenomen in een Google Lighthouse scan.

Door developer.

☑ GZIP / Brotli compressie

Gebruik deze compressietechnieken om de grootte van JavaScript bestanden drastisch te beperken.

Door developer.

Media

☑ Optimaliseer visuals

Het optimaliseren van afbeeldingen is een 3-traps raket: Als eerste bepaal je de maximale breedte/hoogte waarin de afbeelding op een (responsive) site te zien is. Verklein je afbeelding naar die grootte zodat er geen onnodige pixel-data wordt geladen.

Als tweede stap kies je het juiste format voor je afbeelding. Moet de afbeelding een transparante achtergrond hebben? Kies dan een PNG. Is de afbeelding een volvlak foto? Kies dan ten alle tijden een JPG. Gebruik Google Lighthouse om te detecteren welke afbeeldingen kunnen profiteren van nieuwe formats als JPEG2000m, JPEG XR en WebP.

Als derde minificeer je de afbeelding met een tool als TinyPNG/TinyJPG of Kraken. Het is ook mogelijk om dergelijke compressie in je CMS in te bouwen.

Door owner en developer.

☑ Lazy loading

Gebruik lazy loading om afbeeldingen die niet direct zichtbaar zijn pas in te laden wanneer de gebruiker deze nodig heeft.

Door developer.

☑ Responsive afbeeldingen

Genereer verschillende maten van een afbeelding (relatief aan de schermbreedte van gebruikers). Gebruik technieken als srcset of picture om verschillende groottes van de afbeelding in te laden.

Door developer.

☑ Gebruik een CDN

Door het gebruik van een CDN worden afbeeldingen, video’s en andere statische bestanden van een server in de buurt van je publiek ingeladen. Daarnaast worden alle statische bestanden dan met een goede caching-policy vanaf het zelfde domein ingeladen. Veel CDN services kunnen assisteren met het optimaliseren van afbeeldingen en video’s.

Door owner en developer.

Wil je een performance check laten doen? Neem contact op via 076 204 30 46 of mail contact@bluebirdday.nl.