Engineer

WebP

et rasterbildeformat med komprimering med og uten tap

Hva det er

WebP, skrevet med stor W og stor P i løpende tekst og på disk bare en `.webp` -fil, er et rasterbildeformat utgitt av Google, lagret i filer med endelsen `.webp` og servert med den registrerte medietypen `image/webp` . Ett format dekker jobber som før krevde tre: fotografier, som ble sendt ut som JPEG; flat grafikk med skarpe kanter og gjennomsiktighet, som ble sendt ut som PNG; og korte animasjoner i sløyfe, som ble sendt ut som GIF.

Innvendig er en WebP-fil en RIFF-beholder, den samme oppdelte strukturen som WAV-lyd bruker. Den støtter to bildebitstrømmer. En `VP8` -del bærer data med tap, kodet med nøkkelbildemetoden fra videokodeken VP8: koderen forutsier hver blokk med piksler fra naboer den allerede har dekodet, og lagrer bare forskjellen. En `VP8L` -del bærer tapsfrie data, fra en egen koder skrevet for bilder som må komme tilbake bit for bit. En `VP8X` -header markerer den utvidede formen, som kan legge til en alfakanal, animasjonsbilder, en ICC-fargeprofil og metadata i EXIF eller XMP. Google dokumenterte formatet sammen med sitt eget bibliotek, og det er siden publisert som RFC 9649, en informativ RFC fra november 2024.

Den utvidede formen er det som gjør formatet til mer enn en mindre JPEG. Et bilde med tap kan ha en egen alfakanal, så et fotografi med en myk, gjennomsiktig kant må ikke lenger være en tung tapsfri fil. Animasjon ligger i den samme beholderen, bilde for bilde, og hvert bilde er med eller uten tap etter hva som passer den.

Bildebitstrømmene har sine egne grenser. Hver side av et kodet bilde eller et animasjonsbilde lagres i 14 bit, så den stopper på 16 383 piksler for data med tap og 16 384 for tapsfrie data. Den utvidede formen kan plassere animasjonsbilder

på et større lerret, og en hel fil holder seg rett under 4 GiB. Hver kanal lagres i 8 bit, noe som passer bilder ment for en skjerm, men mister den ekstra dybden i en original på 16 bit.

Bruk begrepet i en beslutning

Bruk «Hva er WebP?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Grensesnitt som laster raskt, fungerer på mobilen og oppfører seg likt i alle nettlesere kundene dine måtte bruke.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Styrte fullstack-utviklingen av GIS-kart — fullstack-kartutvikling, fra database til nettleser

Styrte fullstack-utviklingen av GIS-kart Ta kontakt →

Hvorfor det er viktig

Bilder er som regel de tyngste bytene på en side, og formatet avgjør hvor mange det blir. Googles egen sammenligning plasserer tapsfri WebP 26 % under PNG, og WebP med tap 25–34 % under JPEG ved lik SSIM, et mål på strukturell likhet mellom to bilder. En WebP med tap og alfakanal blir typisk en tredjedel av PNG-filen den erstatter. Færre byte når skjermen tidligere på en treg forbindelse, koster mindre å servere og gir plass i budsjettet til siden for det leseren kom for.

Valget et team står overfor, handler mindre om WebP skal brukes, og mer om hvilket format som skal sendes ut for hvilket bilde, og om det skal sendes ut mer enn ett. Der slutter publiserte gjennomsnitt å hjelpe. De er gjennomsnitt over et testkorpus, mens filene på et virkelig nettsted er en logo, en håndfull fotografier og noen skjermbilder, og rangeringen kan falle ulikt ut for hver av dem. AVIF, det nyere formatet, er den vanlige utfordreren, og på papiret komprimerer det hardere. På et gitt nettsted, i størrelsene det nettstedet viser, kan det tape.

Det andre valget er hvor konverteringen skjer. Å konvertere for hånd én gang etterlater en mappe med filer som glir bort fra kildene sine. Å konvertere ved bygging, fra en original som holdes i repoet, gjør hver publiserte størrelse reproduserbar og gjør det billig å endre kvalitetsinnstillingen, eller formatet, senere.

Det tredje er alt utenfor nettleseren. Adressen til et bilde reiser: inn i feeder, e-post, forhåndsvisninger i chat, kontordokumenter og bildeprogrammer, og hver av dem avgjør selv hvilke formater den leser. Et nettsted som flytter hvert bilde

over til WebP, får vite hvilke mottakere som godtar det, fra den første leseren som ser et ødelagt bilde, med mindre det sjekker hver vei først.

Slik ser det ut i praksis

Kolofonsiden oppgir bildene på dette nettstedet som WebP og PNG, skalert ved bygging. Malene viser hva det betyr. Fotografier i sideinnholdet, avatarer og merkevaresymbolet går alle gjennom én felles delmal (*partial*) som skalerer originalen til flere bredder, hver kodet som WebP, og skriver ett enkelt `img` -element med `srcset` , så nettleseren velger bredden, og nettstedet allerede har valgt formatet.

Det formatet er en måling og ikke en standardinnstilling. Hugo-versjonen nettstedet bruker, kan kode AVIF, så å servere begge gjennom et `picture` -element var et alternativ. I sammenligningen dokumentert 17. august 2026, i størrelsene nettstedet da viste, ble AVIF 30–36 % større enn WebP for hjernemerket og 3–7 % større for avatarene; det vant bare på portrettet, med 13 %, og så vidt på kontorfotografiene, med 3–4 %. Et `picture` -element velger kilde etter rekkefølge og etter hva nettleseren støtter, aldri etter størrelse, så å sette AVIF først ville ha sendt hver moderne nettleser omtrent en tredjedel flere byte for logoen på hver side. Med bare ett format verdt å sende ut bruker malen `img` med `srcset` og ikke noe `picture` i det hele tatt.

Den samme delmalen bærer en lærdom til. Hugo slipper ikke et bilde uendret gjennom når det blir bedt om å skalere det til bredden det allerede har: det koder bildet på nytt, noe som gir en ny publisert fil og, for en original med tap, enda en runde med tap. Når en ønsket bredde er lik originalens, peker delmalen derfor direkte på originalen som standard. Kallere som uttrykkelig ber om en avledet fil for hver størrelse, koder det fortsatt på nytt; photo-shortcoden velger den veien for å unngå å publisere en ekstra kopi av en master som allerede serveres via assets-monteringen.

WebP stopper ved grensen for hva annen programvare leser. RSS-feeden peker `image` -elementet sitt mot en PNG, fordi RSS 2.0 bare godtar GIF, JPEG og PNG der og begrenser bildet til 144 ganger 400 piksler: delingskortet i WebP på 1 280 piksler utløste en advarsel fra feedvalidatoren til W3C, og apple-touch-ikonet på 144 piksler, en PNG med nøyaktig den største tillatte bredden, tok plassen. Skjermbildene som det engelske og det danske manifestet gir til installasjonsdialogen i Chromium, er derimot WebP, ett bredt og ett smalt opptak for hvert språk, og en lint-oppgave feiler når det registrerte

genereringstidspunktet deres ligger før en committet endring i gjengivelsesgrunnlaget sjekken følger; endringer som ikke er committet, er utenfor den sjekken.

Hvor det går galt

En konvertert JPEG mister detaljer to ganger. Å gjøre en eksisterende JPEG om til WebP med tap dekoder én generasjon med tap og koder en ny, og artefaktene fra begge overlever. At Hugo koder på nytt når et bilde skaleres til samme bredde, er en stillere form av den samme fellen. Hold en tapsfri original eller en original i full kvalitet i repoet, og kod hver publiserte fil fra den.

Fargene blir matte etter konverteringen. Referansekoderen `cwebp` kopierer ingen metadata med mindre den blir bedt om det, så ICC-profilen til et fotografi med bredt fargeområde blir liggende igjen, og nettleseren behandler pikslene som sRGB. Bruk `-metadata icc`, eller konverter bildet til sRGB før kodingen.

En feedleser viser et ødelagt bilde. Programvare som ble skrevet før formatet, eller som holdes til en eldre spesifikasjon, behandler en WebP som en ukjent fil. RSS 2.0 er det tydeligste tilfellet: `image`-elementet nevner tre formater, og WebP er utenfor dem. Følg hvert sted adressen til et bilde reiser, og lever PNG eller JPEG der mottakeren krever det.

Et langt skjermbilde stopper koderen. Et helsides skjermbilde av en lang artikkel passerer lett 16 383 piksler i høyden. Koderen i libwebp begrenser hver side til den størrelsen, selv om den tapsfrie bitstrømmen kan representere én piksel til. Del opp opptaket, eller skaler det ned først.

Den største filen vinner `picture`-elementet. Å sette et nyere format først sender det til hver nettleser som kan dekode det, enten det er mindre for akkurat det bildet eller ikke. Mål hver kandidat i størrelsene siden viser, før en kilde legges til, slik dette nettstedet gjorde med AVIF.

Les mer

- WebP — Googles hjemmeside for formatet: kompresjonsstudien bak de publiserte prosentene, biblioteket libwebp og verktøyene som følger med.
- RFC 9649: WebP Image Format — beholderen, bitstrømmene med og uten tap, utvidede funksjoner og størrelsesgrensene, som en informativ RFC.

- cwebp — hvert valg i referansekoderen, også `-q`, `-lossless` og `-metadata`.
- Hugo: Image processing — hvordan et Hugo-nettsted skalerer og konverterer bilder, også til WebP, ved bygging.
- HTML Standard: Images — hvordan `srcset`, `sizes` og `picture` lar nettleseren velge mellom kandidatfiler.
- RSS 2.0 Specification — `image`-elementet, de tre formatene det godtar, og størrelsesgrensene.

Ordliste

Velg ut fra deres faktiske rammer

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er WebP?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/webp/