Engineer

WebP

ett rasterformat med destruktiv och förlustfri komprimering

Vad det är

WebP är ett rasterformat för bilder som Google har publicerat. Filerna har ändelsen `.webp` och serveras med den registrerade medietypen `image/webp`. Ett och samma format täcker jobb som förut krävde tre: fotografier, som skickades som JPEG, platt grafik med skarpa kanter och genomskinlighet, som skickades som PNG, och korta loopar, som skickades som GIF. I löptext skrivs namnet med stort W och stort P, och på disken är det helt enkelt filen med `.webp`.

Invändigt är en WebP-fil en RIFF-behållare, samma uppdelning i block (chunks) som ljudfiler i WAV använder. Den stöder två bildbitströmmar. Ett `VP8`-block bär destruktivt komprimerade data, kodade med nyckelbildsmetoden i videokodeken VP8: kodaren förutsäger varje block av pixlar utifrån grannar som den redan har avkodat och sparar bara skillnaden. Ett `VP8L`-block bär förlustfria data, från en separat kodare som är skriven för bilder som måste komma tillbaka bit för bit. Ett `VP8X`-huvud markerar den utökade formen, som kan lägga till en alfakanal, animerade bildrutor, en ICC-färgprofil och metadata i EXIF eller XMP. Google dokumenterade formatet tillsammans med sitt eget bibliotek, och det har sedan publicerats som RFC 9649, en informativ RFC från november 2024.

Den utökade formen är det som gör formatet till mer än en mindre JPEG. En destruktivt komprimerad bild kan ha en separat alfakanal, så ett fotografi med en mjuk, genomskinlig kant behöver inte längre vara en tung förlustfri fil. Animationer ligger i samma behållare, bildruta för bildruta, och varje bildruta är destruktivt eller förlustfritt komprimerad efter vad som passar den.

Bildbitströmmarna har sina egna gränser. Varje sida av en kodad bild eller bildruta lagras med 14 bitar, så den når som mest 16 383 pixlar för destruktiva data och 16 384 för förlustfria. Den utökade formen kan placera bildrutor på en

större bildyta, och en hel fil håller sig strax under 4 GiB. Varje kanal lagras med 8 bitar, vilket passar bilder som är avsedda för en skärm och tappar det extra djupet i ett original med 16 bitar.

Använd begreppet i ett beslut

Använd ”Vad är WebP?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Gränssnitt som laddar snabbt, fungerar i mobilen och beter sig likadant i alla webbläsare dina kunder råkar använda.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Styrde full stack-utvecklingen av GIS-kartor — kartutveckling i full stack, från databas till webbläsare

Styrde full stack-utvecklingen av GIS-kartor Hör av er →

Varför det spelar roll

Bilderna är oftast de tyngsta byten på en sida, och formatet avgör hur många de blir. Den jämförelse som Google själv har gjort lägger förlustfri WebP 26 % under PNG, och destruktiv WebP 25–34 % under JPEG vid samma SSIM, ett mått på strukturell likhet mellan två bilder. En destruktiv WebP med alfakanal hamnar typiskt på en tredjedel av den PNG-fil som den tar över efter. Färre byte når skärmen snabbare på en långsam anslutning, kostar mindre att servera och lämnar plats i sidans budget för det som läsaren kom för.

Beslutet som ett team står inför gäller inte så mycket om WebP ska användas som vilket format ni ska leverera för vilken bild, och om ni ska leverera mer än ett. Publicerade genomsnitt slutar hjälpa där. De är genomsnitt över en testsamling, medan filerna på en verklig webbplats är en logotyp, en handfull fotografier och några skärmbilder, och rangordningen kan bli en annan för var och en. AVIF, det nyare formatet, är den vanliga rivalen, och på papperet komprimerar det hårdare. På en given webbplats, i de storlekar som just den webbplatsen visar, kan det förlora.

Det andra beslutet gäller var konverteringen sker. Att konvertera för hand en gång lämnar en mapp med filer som glider ifrån sina källor. Att konvertera när webbplatsen byggs, från ett original som hålls i repot, gör varje publicerad storlek reproducerbar och gör det billigt att ändra kvalitetsinställningen, eller formatet, längre fram.

Det tredje är allt utanför webbläsaren. En bilds adress vandrar vidare: in i webbflöden, e-post, förhandsvisningar i chattar, kontorsdokument och bildredigerare, och var och en av dem bestämmer själv vilka format den läser. En webbplats som flyttar varje bild till WebP får veta vilka mottagare som godtar formatet av den första läsare som ser en trasig bild, om den inte har kontrollerat varje väg först.

Så ser det ut i praktiken

Kolofonen anger att webbplatsens bilder är WebP- och PNG-bilder, skalade när webbplatsen byggs. Mallarna visar vad det innebär. Fotografier i sidinnehållet, avatarer och varumärkessymbolen passerar alla genom en gemensam partial, en återanvändbar mallbit, som skalar om originalet till flera bredder, var och en kodad som WebP, och skriver ett enda `img` -element med en `srcset` , så att webbläsaren väljer bredden medan webbplatsen redan har valt formatet.

Det formatet är ett mätresultat snarare än ett standardval. Den version av Hugo som webbplatsen använder kan koda AVIF, så att leverera båda genom ett `picture` -element var ett alternativ. I jämförelsen som dokumenterades den 17 augusti 2026, vid de storlekar som webbplatsen då visade, blev AVIF 30–36 % större än WebP för hjärnan i logotypen och 3–7 % större för avatarerna. Det vann bara på porträttet, med 13 %, och knappt på kontorsfotografierna, med 3–4 %. Ett `picture` -element väljer källa efter ordning och efter vad webbläsaren stöder, aldrig efter storlek, så om AVIF hade stått först hade varje modern webbläsare fått ungefär en tredjedel fler byte för logotypen på varje sida. Med ett enda format som är värt att leverera använder mallen `img` med `srcset` och inget `picture` alls.

Samma partial bär på en lärdom till. Hugo släpper inte igenom en bild oförändrad när den ombeds skala om den till den bredd som den redan har, utan kodar om den. Det ger en andra publicerad fil och, för ett destruktivt komprimerat original, ännu ett destruktivt varv. När en begärd bredd är lika med originalets pekar partialen därför direkt på originalet som standard. Anropare som uttryckligen begär härledda filer för varje storlek kodar fortfarande om bilden; photo-shortcoden väljer den vägen för att slippa publicera ännu en kopia av ett original som redan serveras via assets-monteringen.

WebP tar slut vid gränsen för vad annan programvara läser. RSS-flödet låter sitt `image` -element peka på en PNG, eftersom RSS 2.0 bara godtar GIF, JPEG och PNG där och begränsar bilden till 144 × 400 pixlar: delningsbilden i WebP, 1 280

pixlar bred, gav en varning i flödesvalidatorn från W3C, och apple-touch-ikonen på 144 pixlar, en PNG med exakt den största tillåtna bredden, tog dess plats. De skärmbilder som det engelska och det danska manifestet ger till installationsdialogen i Chromium är däremot WebP, en bred och en smal för vart och ett, och en lintuppgift fallerar när deras registrerade genereringstidpunkt ligger före en committad ändring i de renderingsunderlag som kontrollen följer; ocommittade ändringar omfattas inte.

Var det går fel

En konverterad JPEG tappar detaljer två gånger. Att göra om en befintlig JPEG till destruktiv WebP avkodar en destruktiv generation och kodar en ny, och artefakterna från båda överlever. Att Hugo kodar om en bild som skalas till sin egen bredd är en tystare variant av samma fälla. Behåll ett förlustfritt original, eller ett i full kvalitet, i repot och koda varje publicerad fil från det.

Färgerna blir matta efter konverteringen. `cwebp`, referenskodaren, kopierar inga metadata om den inte blir tillsagd, så ICC-profilen i ett fotografi med brett färgomfång lämnas kvar och webbläsaren behandlar pixlarna som sRGB. Ange `-metadata icc`, eller konvertera bilden till sRGB före kodningen.

En flödesläsare visar en trasig bild. Programvara som skrevs före formatet, eller som håller sig till en äldre specifikation, behandlar en WebP som en okänd fil. RSS 2.0 är det tydligaste fallet: dess `image`-element nämner tre format, och WebP är inte ett av dem. Följ varje ställe dit en bilds adress vandrar, och servera PNG eller JPEG där mottagaren kräver det.

En lång skärmbild stoppar kodaren. En helsidesbild av en lång artikel passerar lätt 16 383 pixlar i höjd. Kodaren i libwebp begränsar varje sida till den storleken, trots att den förlustfria bitströmmen kan representera ytterligare en pixel. Dela upp bilden, eller skala ned den först.

Den större filen vinner i `picture`-elementet. Att lista ett nyare format först skickar det till varje webbläsare som kan avkoda det, oavsett om det är mindre för just den bilden. Mät varje kandidat i de storlekar som sidan visar innan du lägger till en källa, som den här webbplatsen gjorde med AVIF.

Läs mer

- WebP — formatets hemvist hos Google: komprimeringsstudien bakom de publicerade procentsatserna, biblioteket libwebp och dess verktyg.
- RFC 9649: WebP Image Format — behållaren, de destruktiva och förlustfria bitströmmarna, utökade funktioner och storleksgränserna, som en informativ RFC.
- cwebp — varje alternativ i referenskodaren, bl.a. `-q`, `-lossless` och `-metadata`.
- Hugo: Image processing — hur en webbplats byggd med Hugo skalar om och konverterar bilder, WebP inräknat, under bygget.
- HTML Standard: Images — hur `srcset`, `sizes` och `picture` låter webbläsaren välja bland kandidatfiler.
- RSS 2.0 Specification — `image`-elementet, de tre format som det godtar och dess storleksgränser.

Ordlista

Välj utifrån era faktiska förutsättningar

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är WebP?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

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

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/webp/