Engineer

Statisk nettstedsgenerator

⚡ et program som skriver ut hver side på forhånd

Hva det er

En statisk nettstedsgenerator er et byggverktøy som tar en mappe med kildefiler, som regel sider skrevet i Markdown og et sett med maler, og gjør dem om til et ferdig nettsted som bare består av filer: HTML, stilark, skript og bilder som en webserver sender nøyaktig slik de ligger. Den kalles også en generator for statiske nettsteder, og på engelsk forkortes den ofte til SSG. Ordet «statisk» beskriver hvordan resultatet leveres, med de samme bytene til alle som ber om en adresse, og sier ingenting om hvordan siden oppfører seg når den først er kommet fram til nettleseren.

Motstykket er den dynamiske måten å gjøre det på, med et publiseringssystem eller en webapplikasjon der et program på serveren setter sammen hver side idet den blir bedt om, som regel ved å spørre en database og fylle ut maler. En generator kjører den samme typen maler over den samme typen innhold, men én gang per endring og ikke én gang per besøk, og legger resultatet i en mappe som en vanlig webserver, en lagringsbøtte eller et CDN kan levere uten å vite hvordan sidene ble laget.

Hugo er én slik generator. Dokumentasjonen presenterer den som en statisk nettstedsgenerator skrevet i Go, optimalisert for fart og laget for fleksibilitet, som gjengir et helt nettsted på sekunder og ofte raskere. Den vanlige arbeidsrytmen er å skrive med den innebygde webserveren i gang, som bygger på nytt ved hver endring og laster de åpne sidene i nettleseren inn igjen, og så bygge nettstedet og produksjonssette det hos en vert:

```
hugo server     # build, serve, rebuild and reload on every save
hugo --minify   # write the finished site into public/
```

Bruk begrepet i en beslutning

Bruk «Hva er en statisk nettstedsgenerator?» 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

Valget handler om når arbeidet gjøres. En generator flytter det fra forespørselen til byggingen, og nesten alt annet følger av det.

Å sende en fil er omtrent det billigste en webserver gjør, så et generert nettsted svarer raskt og kan kjøre på en svært liten maskin, og fordi hver besøkende på en adresse får de samme bytene, er hurtigbufring enkelt. Når det ikke kjører applikasjonskode eller en database bak sidene, er det langt mindre å sikre og oppdatere. Resultatet er en mappe, så å flytte til en annen vert betyr å kopiere filer, og kilden ligger i versjonskontroll, der hver endring på hver side kan gjennomgås og spores.

Kostnadene kommer fra det samme stedet. Hver endring, også en rettet skrivefeil, betyr en ny bygging og en ny produksjonssetting, så byggefarten betyr mer jo flere sider det blir. Alt som avhenger av hvem som spør, f.eks. søk, kommentarer, innlogging eller et skjema som lagrer det det får, trenger kode i nettleseren, en egen tjeneste eller en måte å bli avgjort ved bygging på, fordi det ikke finnes noe program på serveren som kan gjøre det. Tiden står stille mellom to bygginger. Hugo utelater som standard en side med en dato fram i tid, så en side planlagt til mandag dukker opp med den første byggingen på eller etter mandag. Selv videresendinger skifter form. Når det ikke finnes noe program som kan sende en HTTP-videresending, er standarden i Hugo for en flyttet side en liten HTML-fil på den gamle adressen som sender nettleseren videre, en løsning dokumentasjonen beskriver som flyttbar mellom alle vertsleverandører.

Slik ser det ut i praksis

Dette nettstedet er bygd med en. Kolofonen navngir Hugo som den statiske nettstedsgeneratoren og gir den æren for malene, taksonomiene, hvert språk og feedene. Om det som når nettleseren, sier den at HTML-en er skrevet ferdig før

noen ber om den: hver side er en fil på disk og ingen mal som settes sammen ved hver forespørsel. Selv utseendet avgjøres ved bygging, siden veggen i det lyse temaet trekkes mens siden bygges, og nettstedet kles på nytt hver gang det publiseres. Det finnes ingen ekstern byggetjeneste. Kvalitetskontrollen kjører på maskinen nettstedet skrives på, hver diagnose teller som en feil, og i oktober 2026 bygde nettstedet 2 268 sider. De leveres fra en server med én kjerne og 512 MB minne. Der leser webserveren og de tre andre tjenestene som svarer for nettstedet, de samme filene Hugo skrev, så én produksjonssetting når alle fire.

Porteføljeoppføringen om å bygge dette nettstedet i Hugo dokumenterer implementasjonsstadiet i august 2026 og gir begrunnelsen. Tre språk, ingen separat fakturert applikasjonstjeneste og ingen besøkende gitt videre til tredjeparter utelukket nesten alle publiseringssystemer, fordi en backend som kjører, er noe som må sikres, patches, betales for og forklares på en personvernside. Nettstedet hadde da 108 malfiler, 53 av dem partials, publiserte hver side i fire representasjoner fra ett innholdstre og bygde det treet sju ganger til for fokuserte underdomener ved å peke generatoren mot en annen innholdsmappe. Oppføringen beskriver avveiningen ved å legge til funksjonalitet ved bygging og ikke i nettleseren: utover det dokumenterte, avgrensede JavaScript-unntaket for et porteføljefilter måtte interaktive funksjoner løses ved bygging eller utelates, og derfor ble en søkeindeks kostnadsberegnet og lagt på is i stedet for levert.

Hvor det går galt

En slettet side som blir liggende på nett. En side som er fjernet fra kilden, leveres fortsatt på den gamle adressen, og søkemotorene fortsetter å finne den. Hugo tømmer ikke utdatamappen før en bygging: filer som finnes, blir skrevet over, men aldri slettet, så `public/` kan beholde sider fra tidligere bygginger, bl.a. utkast og utløpte sider, og alt som kopierer den mappen til en server, kopierer dem med. Hurtigbufferen i `resources/` har en slektning av det samme problemet: designreglene for dette nettstedet advarer om at den kan levere en gammel CSS-pakke når bare kilde-CSS-en er endret. Bygg inn i en tom mappe, med flagget `--cleanDestinationDir` i Hugo eller et eget slettesteg. Byggingen av dette nettstedet fjerner både `public/` og `resources/` før den begynner.

En ferdig side som aldri dukker opp. Et innlegg blir skrevet, sjekket inn og produksjonssatt, byggingen melder suksess, og nettstedet har ingen side for det. Som standard utelater Hugo innhold merket som utkast, innhold der datoen eller publiseringsdatoen ligger fram i tid, og innhold der utløpsdatoen er passert, og

ingenting av det teller som en feil. En dato skrevet inn én dag fram i tid er nok. Sjekk de fire feltene først, hold `--buildDrafts`, `--buildFuture` og `--buildExpired` til forhåndsvisninger, og sørg for en bygging etter datoen når en side er planlagt.

Et sitemap der alle sidene er endret i samme øyeblikk. Nesten hver adresse får det samme tidspunktet for siste endring, noe som forteller en søkerobot at hele nettstedet er endret, og ingenting om hvilke sider som faktisk er det. En generator må hente endringsdatoen for hver side et sted fra, og filens eget tidsstempel er feil kilde når et steg skriver om filer der innholdet er det samme. Konfigurasjonen for dette nettstedet registrerer at synkroniseringssteget skriver om alle de 189 innholdsfilene ved hver kjøring, så 172 av de 176 adressene i sitemapet delte ett tidsstempel. Hent datoen fra versjonskontrollen i stedet. Hugo leser datoen for den siste Git-revisjonen av en fil når `enableGitInfo` er slått på, og en ny synkronisering som gir identiske byte, lager ingen commit, så en synkronisering etter en måned med endringer rørte 8 av 186 filer.

Filer som går ut uten at noen mente å publisere dem. Alle som gjetter navnet, kan hente en fil som lister opp andre filer. En generator publiserer det som ligger i mappene den leser, minus det en unntaksregel fjerner, og en regel som ikke når hver mappe, lar resten ligge åpent. Konfigurasjonen for dette nettstedet registrerer at listen over filer som skulle overses på hele nettstedet, ikke nådde en modulmontering: byggingen publiserte alle seks `.DS_Store`-filene som Finder i macOS hadde lagt igjen under icons/, images/ og webmanifest/, på hver vert, og hver av dem listet navnene ved siden av seg, så den i images/ navnga en mappe nettstedet holder tilbake. Et filter som manglet ett mellomrom etter `!`, utelukket en gang i stedet hvert bilde i en montering. Legg unntakene på hver montering som publiserer, og sjekk deretter det bygde resultatet for filer som ikke hører hjemme der. To av kontrollene for dette nettstedet stopper en bygging som leverer en slik fil.

Les mer

- Hugo: Introduction — hva Hugo er, hva den brukes til, og hvordan rytmen med å skrive, bygge og produksjonssette går.
- Hugo: Basic usage — å bygge og levere et nettsted, hvorfor `public/` beholder gamle filer, og hvilket innhold en bygging utelater.
- Hugo: Configure front matter — hvordan Hugo velger datoene for en side, også symbolene `:git` og `:fileModTime`.
- Hugo: Configure modules — monteringer, `files`-filteret hver av dem tar, og regelen om at en definert montering fjerner standardmonteringen.

- Hugo: URL management — aliaser, og videresendingsfilene en statisk vert leverer i stedet for en videresending fra serveren.

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 en statisk nettstedsgenerator?» 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/ssg/