**Engineer**

# Statisk webbplatsgenerator

ett program som skriver ut varje sida i förväg

## Vad det är

En statisk webbplatsgenerator är ett byggverktyg som tar en mapp med källfiler, oftast sidor skrivna i Markdown och en uppsättning mallar, och gör om dem till en färdig webbplats som bara består av filer: HTML, stilmallar, skript och bilder som en webbserver skickar precis som de är. Begreppet förkortas ofta SSG, och i löpande text räcker det ofta med webbplatsgeneratorn. Ordet ”statisk” beskriver hur resultatet levereras, samma byte till alla som frågar efter en adress, inte hur en sida beter sig när den väl har nått webbläsaren.

Motsatsen är det dynamiska arbetssättet i ett system för innehållshantering eller en webbapplikation, där ett program på servern sätter ihop varje sida när den efterfrågas, oftast genom att fråga en databas och fylla i mallar. En generator kör samma slags mallar över samma slags innehåll, men en gång per ändring i stället för en gång per besök, och lämnar resultatet i en mapp som en vanlig webbserver, en lagringshink eller ett CDN kan leverera utan att veta hur sidorna kom till.

Hugo är en sådan generator. Dess dokumentation presenterar den som en statisk webbplatsgenerator skriven i Go, optimerad för hastighet och utformad för flexibilitet, som renderar en hel webbplats på några sekunder, ofta kortare tid än så. Det vanliga arbetsflödet är att skriva med den inbyggda webbservern igång, som bygger om vid varje ändring och laddar om de öppna sidorna i webbläsaren, och sedan bygga webbplatsen och driftsätta den hos en värd:

```
hugo server     # bygg, servera, bygg om och ladda om vid varje sparning
hugo --minify   # skriv den färdiga webbplatsen till public/
```

Använd begreppet i ett beslut

Använd ”Vad är en statisk webbplatsgenerator?” 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

Valet handlar om när arbetet görs. En generator flyttar det från tidpunkten för förfrågan till byggtiden, och nästan allt som följer kommer därifrån.

Att skicka en fil är ungefär det billigaste en webbserver gör, så en genererad webbplats svarar snabbt och kan köras på en mycket liten maskin, och eftersom varje besökare på en adress får samma byte blir cache enkel. Utan applikationskod eller en databas som körs bakom sidorna finns det mycket mindre att skydda och patcha. Resultatet är en mapp, så att flytta till en annan värd innebär att kopiera filer, och källan finns i versionshantering, där varje ändring av varje sida kan granskas och spåras.

Kostnaderna kommer från samma ställe. Varje ändring, också ett rättat stavfel, kräver ett nytt bygge och en ny driftsättning, så byggets hastighet betyder mer ju fler sidorna blir. Allt som beror på vem som frågar, som sökning, kommentarer, inloggning eller ett formulär som sparar det det tar emot, behöver kod i webbläsaren, en separat tjänst eller ett sätt att avgöras vid byggtid, eftersom det inte finns något program på servern som kan göra det. Tiden står still mellan byggena: Hugo utelämnar som standard en sida vars datum ligger i framtiden, så en sida som är planerad till måndag dyker upp med det första bygget på eller efter måndagen. Även omdirigeringar byter form. Utan ett program som skickar en omdirigering i HTTP är Hugos standard för en flyttad sida en liten HTML-fil på den gamla adressen som skickar webbläsaren vidare, ett arbetssätt som dokumentationen kallar portabelt mellan alla värdleverantörer.

## Så ser det ut i praktiken

Den här webbplatsen byggs med en. Kolofonen nämner Hugo som den statiska webbplatsgeneratorn och tillskriver den mallarna, taxonomierna, varje språk och flödena. Om det som når webbläsaren säger den att HTML-koden skrivs ut i sin

helhet innan någon frågar efter den: varje sida är en fil på disk, inte en mall som sätts ihop per förfrågan. Till och med utseendet avgörs vid byggtid, eftersom det ljusa temats vägg lottas medan sidan byggs och webbplatsen kläs på nytt varje gång den publiceras. Bygget stannar hemma: kvalitetsgrinden körs utan extern byggtjänst på den maskin som webbplatsen skrivs på, varje diagnos räknas som ett fel, och i oktober 2026 byggde webbplatsen 2 268 sidor. De levereras från en server med en kärna och 512 MB minne, där webbservern och de tre andra tjänsterna hämtar allt ur en och samma mapp, och en enda driftsättning når därför alla fyra.

Portföljposten om att bygga den här webbplatsen i Hugo dokumenterar genomförandeskedet i augusti 2026 och ger skälen. Tre språk, ingen separat fakturerad applikationstjänst och inga besökare som lämnas ut till tredje parter uteslöt nästan varje system för innehållshantering, eftersom en backend som körs är något som ska skyddas, patchas, betalas för och förklaras på en integritetssida. Webbplatsen hade då 108 mallfiler, varav 53 var partials, publicerade varje sida i fyra representationer från ett och samma innehållsträd och byggde det trädet sju gånger till för fokuserade underdomäner, genom att peka generatorn mot en annan innehållskatalog. Posten beskriver avvägningen med att lägga till funktioner vid byggtid i stället för i webbläsaren: utöver det dokumenterade, avgränsade JavaScript-undantaget för ett portföljfilter behövde interaktiva funktioner lösas vid byggtid eller utelämnas, och det var därför ett sökindex kostnadsberäknades och parkerades i stället för att levereras.

## Var det går fel

**En raderad sida ligger kvar på nätet.** En sida som tagits bort ur källan levereras fortfarande på sin gamla adress, och sökmotorer fortsätter att hitta den. Hugo tömmer inte sin utdatamapp före ett bygge: befintliga filer skrivs över men raderas aldrig, så `public/` kan behålla sidor från tidigare byggen, utkast och utgångna sidor bland dem, och allt som kopierar den mappen till en server kopierar dem också. Cachen i `resources/` har en släkting till problemet: den här webbplatsens designregler varnar för att den kan leverera ett gammalt CSS-paket när bara CSS-källan har ändrats. Bygg till en ren mapp, med Hugos flagga `--cleanDestinationDir` eller ett eget raderingssteg; den här webbplatsens bygge tar bort både `public/` och `resources/` innan det börjar.

**En färdig sida dyker aldrig upp.** Ett inlägg skrivs, checkas in och driftsätts, bygget rapporterar att allt gick bra, och webbplatsen har ingen sida för det. Som standard utelämnar Hugo innehåll som är markerat som utkast, innehåll vars datum eller publiceringsdatum ligger i framtiden och innehåll vars utgångsdatum

har passerat, och inget av det räknas som ett fel. Ett datum som skrivits en dag in i framtiden räcker. Kontrollera de fyra fälten först, spara `--buildDrafts`, `--buildFuture` och `--buildExpired` till förhandsvisningar, och ordna ett bygge efter datumet för en planerad sida.

**Alla sidor påstår att de ändrades på en gång.** Webbplatskartan ger nästan varje adress samma tid för senaste ändring, vilket säger till en sökrobot att hela webbplatsen ändrades och ingenting om vilka sidor som faktiskt gjorde det. En generator måste hämta varje sidas ändringsdatum någonstans ifrån, och filens egen tidsstämpel är fel källa när ett steg skriver om filer vars innehåll förblir detsamma. Den här webbplatsens konfiguration noterar att dess synksteg skriver om alla 189 innehållsfiler vid varje körning, så 172 av de 176 adresserna i webbplatskartan delade en tidsstämpel. Ta datumet från versionshanteringen i stället: Hugo läser datumet för en fils senaste revision i Git när `enableGitInfo` är påslaget, och en ny synk som ger identiska byte blir ingen incheckning, så en ny synk efter en månads redigeringar rörde 8 av 186 filer.

**Filer som ingen tänkt publicera går ut.** Den som gissar namnet kan hämta en fil som listar andra filer. En generator publicerar allt som ligger i de mappar den läser, minus det som en uteslutningsregel tar bort, och en regel som inte når varje mapp lämnar resten exponerad. Den här webbplatsens konfiguration noterar att dess uteslutningslista för hela webbplatsen inte nådde en modulmontering: bygget publicerade alla sex `.DS_Store`-filer som Finder i macOS hade lämnat under icons/, images/ och webmanifest/, på varje värd, och var och en listade namnen bredvid sig, så den i images/ namngav en mapp som webbplatsen håller inne med. Ett filter som saknade ett mellanslag efter sitt `!` uteslöt en gång i stället varje bild i en montering. Uteslut på varje montering som publicerar, och kontrollera sedan det byggda resultatet efter filer som inte hör hemma där; två av den här webbplatsens kontroller fäller ett bygge som levererar en sådan.

## Läs mer

- Hugo: Introduction — vad Hugo är, vad den används till och hur arbetsflödet med skrivning, bygge och driftsättning går till.
- Hugo: Basic usage — att bygga och servera en webbplats, varför `public/` behåller gamla filer och vilket innehåll ett bygge utelämnar.
- Hugo: Configure front matter — hur Hugo väljer en sidas datum, inklusive tokenen `:git` och `:fileModTime`.
- Hugo: Configure modules — monteringar, filtret `files` som var och en tar, och regeln att en definierad montering tar bort standardmonteringen.

- Hugo: URL management — alias, och de omdirigeringsfiler som en statisk värd levererar i stället för en omdirigering från servern.

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 en statisk webbplatsgenerator?” 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/ssg/**