

Engineer

静的サイトジェネレーター

⚡ すべてのページを事前にファイルとして書き出すプログラム

どのようなものか

静的サイトジェネレーターは、ソースファイルのフォルダーを受け取り、ファイルだけで構成された完成済みのウェブサイトを生成するビルドツールです。ソースは通常、Markdownで書いたページと一式のテンプレートで、出力はHTML、スタイルシート、スクリプト、画像になります。ウェブサーバーは、それらをそのまま送信します。SSGと略されることも多く、「静的」という言葉は、あるアドレスを要求する誰に対しても同じバイト列を返すという配信方法を指します。ブラウザーに届いた後のページの振る舞いを指す言葉ではありません。

対照的なのが、コンテンツ管理システムやウェブアプリで使う動的な方式です。こちらは要求があるたびに、通常はデータベースへ問い合わせてテンプレートに内容を埋め込み、サーバー上のプログラムがページを組み立てます。ジェネレーターも同種のコンテンツに同種のテンプレートを適用しますが、訪問のたびではなく変更のたびに一度だけ実行します。出力はフォルダーに残り、通常のウェブサーバー、ストレージバケット、コンテンツ配信ネットワークは、そのページをどう作ったかを知らなくても配信できます。

Hugoは、そのようなジェネレーターの一つです。公式ドキュメントでは、「Goで書かれ、高速動作に向けて最適化され、柔軟性を重視して設計された静的サイトジェネレーター」であり、「サイト全体を数秒、多くの場合はそれより短い時間でレンダリングする」と紹介されています。通常は、組み込みのウェブサーバーを動かしたまま執筆し、変更のたびに再ビルドして、開いているブラウザーのページを再読み込みさせます。その後、サイトをビルドしてホストへデプロイします。

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

概念を判断に生かす

「静的サイトジェネレーターとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 読み込みが速く、スマートフォンでも使え、お客様がどのブラウザーを使っていても同じように動く画面を提供します。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 フルスタックのGIS地図開発を指揮 — DBからブラウザーまでのフルスタック地図開発

フルスタックのGIS地図開発を指揮 お問い合わせ →

なぜ重要なのか

この選択の中心にあるのは、処理をいつ行うかです。ジェネレーターは処理をリクエスト時からビルド時へ移し、その後に生じる違いのほとんどは、この移動に由来します。

ファイルの送信は、ウェブサーバーにとって最も負担の小さい処理の一つです。そのため、生成済みのサイトは素早く応答でき、ごく小さなマシンでも動かせます。同じアドレスへ来る訪問者全員に同じバイト列を返すので、キャッシュも単純です。ページの裏でアプリのコードやデータベースが動かない分、保護や更新が必要な対象も大幅に減ります。出力はフォルダーなので、別のホストへ移すにはファイルをコピーすれば済みます。ソースはバージョン管理下にあり、どのページのどの変更もレビューし、追跡できます。

制約も同じ理由から生まれます。誤字を一つ直すだけでも、再ビルドとデプロイが必要になるので、ページが増えるほどビルド速度が重要になります。検索、コメント、ログイン、受け取った内容を保存するフォームなど、誰が要求しているかに依存する機能には、ブラウザー内のコード、別のサービス、またはビルド時に処理を済ませる方法が必要です。サーバー上には、その処理をするプログラムがないためです。ビルドと次のビルドの間では、時が止まります。Hugoは既定では未来の日付を持つページを含めないため、月曜日に公開予定のページは、月曜日以降の最初のビルドで現れます。リダイレクトも形が変わります。HTTPリダイレクトを送るプログラムがないので、Hugoは既定では、移動前のアドレスに小さなHTMLファイルを置いてブラウザーを転送します。ドキュメントでは、どのホスティング事業者でも使える方式と説明しています。

実際の活用例

このサイトも、この方式で構築しています。クレジットのページには、静的サイトジェネレーターとしてHugoを挙げ、テンプレート、タクソノミー、すべての言語、フィードを担うものとして記載しています。ブラウザーへ届くHTMLは、

要求される前にすべて書き出されています。各ページは、リクエストごとに組み立てるテンプレートではなく、ディスク上のファイルです。見た目さえビルド時に決まります。ライトテーマの背景はサイトのビルド時にランダムに選ばれ、公開するたびにサイトの装いが新しくなります。品質チェックは編集に使うコンピューター上でローカルに実行し、その工程には外部のホスティング型ビルドサービスを使いません。すべての診断をエラーとして扱い、2026年10月には2,268ページをビルドしました。配信元はメモリー512 MBのシングルコアサーバーで、ウェブサーバーと、このサイトの要求に応答するほかの三つのサービスは、同じファイルのフォルダーを読みます。そのため、一度のデプロイで四つすべてを更新できます。

Hugoでこのサイトを構築した実績記事は、2026年8月の実装段階と、その判断理由を記録しています。三つの言語で使え、別料金のアプリケーションサービスを必要とせず、訪問者を第三者へ引き渡さないという条件により、ほぼすべてのコンテンツ管理システムが候補から外れました。バックエンドを稼働させるなら、その保護と更新を行い、運用費を負担し、プライバシーのページでも説明しなければならないためです。当時のサイトには108個のテンプレートファイルがあり、そのうち53個がパーシャルでした。一つのコンテンツツリーから各ページを四つの表現形式で公開し、ジェネレーターへ別のコンテンツディレクトリを指定して、同じツリーをさらに七回、分野別サブドメイン用にビルドしていました。記事は、ブラウザーではなくビルド時に機能を追加するというトレードオフも記録しています。記録されたポートフォリオのフィルター向けの限定的なJavaScriptの例外を除き、対話的な機能はビルド時に実現するか、採用しないかのどちらかでした。そのため、検索インデックスは負担を見積もったうえで、公開せず保留にしました。

問題が起きるところ

削除したページが公開されたままになる。 ソースから削除したページが以前のアドレスで配信され続け、検索エンジンにも見つかり続けます。Hugoはビルド前に出力フォルダーを空にしません。既存のファイルは上書きしますが、削除はしないため、`public/` には以前のビルドのページが残ることがあり、下書きや公開期限切れのページも含まれます。そのフォルダーをサーバーへコピーする処理は、それらも一緒にコピーします。`resources/` キャッシュにも、似た問題があります。このサイトの設計ルールでは、ソースCSSだけを変更した場合に、古いCSSバンドルが使われることがあると注意しています。Hugoの `--cleanDestinationDir` フラグか、独自の削除工程を使い、空のフォルダーへビルドします。このサイトでは、開始前に `public/` と `resources/` の両方を削除します。

完成したページがいつまでも現れない。 記事を書き、コミットしてデプロイし、ビルドは成功を報告したのに、サイトにはそのページがありません。Hugoは既定では、下書きの指定があるコンテンツ、日付または公開日が未来のコンテンツ、有効期限が過ぎたコンテンツを除外し、いずれもエラーにはしません。日付を一日先に打ち間違えるだけで起こります。まず、その四つの項目を確認します。`--buildDrafts`、`--buildFuture`、`--buildExpired`はプレビュー用にとどめ、公開予約したページには、その日付以降にビルドを実行するよう手配します。

全ページが一斉に変更されたと表示される。 サイトマップのほぼすべてのアドレスに同じ最終更新時刻が付きます。これではクローラーにサイト全体が変わったと伝えるだけで、どのページが変わったかは分かりません。ジェネレーターは各ページの更新日をどこかから取得する必要がありますが、内容の変わらないファイルを別の工程が書き直す場合、ファイル自体のタイムスタンプは適切な取得元ではありません。このサイトの設定記録では、同期工程が毎回189個のコンテンツファイルすべてを書き直したため、サイトマップの176アドレスのうち172個に同じタイムスタンプが付いていました。代わりに、バージョン管理から日付を取得します。Hugoは`enableGitInfo`を有効にすると、そのファイルのGitでの最終リビジョンの日付を読みます。同一のバイト列を生成する再同期ではコミットが生じないため、一か月分の編集後に再同期した際には、186個のファイルのうち変更対象は8個でした。

公開するつもりのないファイルまで公開される。 名前を推測した人なら誰でも、ほかのファイルの一覧が入ったファイルを取得できてしまいます。ジェネレーターは、除外ルールに該当するもの以外、読み込むフォルダー内のあらゆるものを公開します。ルールがすべてのフォルダーに届かなければ、残りは露出します。このサイトの設定記録によると、サイト全体の除外リストがモジュールのマウントに適用されず、macOSのFinderがicons/、images/、webmanifest/の下に残した六つの`.DS_Store`ファイルが、すべてのホストで公開されました。それぞれ同じディレクトリ内の名前を列挙しており、images/内のものには、このサイトが公開しないフォルダーの名前もありました。かつては、フィルターの`!`の後に空白が一つ足りないだけで、逆にマウント内の画像がすべて除外されたこともあります。公開を行う各マウントで除外を設定し、生成済みの出力に不要なファイルがないか確認します。このサイトでは、そのようなファイルを配信するビルドを、二つのチェックで失敗させます。

参考資料

- Hugo: Introduction — Hugoの概要と用途、執筆、ビルド、デプロイという作業の流れを説明しています。
- Hugo: Basic usage — サイトのビルドと配信、`public/`に古いファイルが残る理由、ビルドで除外されるコンテンツを説明しています。

- Hugo: Configure front matter — `:git` と `:fileModTime` トークンを含め、Hugoがページの日付を選ぶ方法を説明しています。
- Hugo: Configure modules — マウント、各マウントが受け取る `files` フィルター、マウントを定義すると既定のマウントがなくなるというルールを説明しています。
- Hugo: URL management — エイリアスと、サーバーによるリダイレクトの代わりに静的ホストが配信する転送用ファイルを説明しています。

用語集

実際の制約に合わせて選ぶ

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「静的サイトジェネレーターとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

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

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/ssg/