Build static sites with Kontent

January 17, 2022 • Martina Farkasova • 8 min read • JavaScript

If you're looking to launch a static, secure, and speedy website, you may be considering a static site generator (SSG). In combination with the JAMstack approach and a headless CMS to store your content, you're on the way to success.

Choose your favorite technology and see what tools are available to you out of the box. And if you're still a bit uncertain whether static sites are the way to go, check out the main benefits SSGs have to offer.

Benefits of static site generators

Static site generators (SSG) take the prepared content, typically stored in flat files, apply it against templates, and generate a structure of static HTML files, ready to be delivered to your visitors. When compared to dynamic websites, they come with several advantages:

- Speed – An absence of the database makes the static site much more speedy and easier to load. All the server needs to do is return a file back to the user. There is no need for database queries, no client-server requests to process. Static site generators store a pre-built version of the site that can be delivered nearly instantly.
- Less complexity – To build a dynamic site, you would need HTML, CSS, and JavaScript on the front end, a server-side scripting language, such as PHP or Perl, on the back end, and SQL to run a database. To build a static site, all you need are front end languages and a templating language on top of that.
- Security – Every request for a web page made to a dynamic site causes an application to run and fetch content from a database. This creates a high risk of the site being attacked and the data being stolen. With a static site, security is the job of the web server. A static site using a CDN is practically immune to attacks because even if one web server on the CDN network goes down from an attack, other servers are still available. And since static sites don't process user data, there is no data to be stolen.
- Scalability – Unexpected traffic surges might crash a dynamic site. A static site is much better prepared as delivering static pages consumes very little server resources. Basic static sites with HTML files can be easily scaled up by just increasing the bandwidth.
- SEO – Load time speed is crucial to get ranked higher by search engines. And your static site will always be faster than dynamically generated websites. Also, static content is easier to parse by the web crawlers.
- Hosting and cost – HTML files for your static site can be served anywhere, scaled and migrated as needed. Plus, the files require less space making the hosting of static websites cheaper to that of dynamic sites.

But of course, running a static site has its disadvantages as well. Serving static pre-rendered content usually means that you cannot tailor the experience on your site for each and every user. If serving real-time data and dynamic experience is a must for your site, dynamic sites are the way to go.

Combining the best of both worlds

Static site generators help you create lightweight, fast-performing, and secure websites. Using an SSG alone, however, has its limitations.

You might be OK with editing text files in Markdown format every time something on your site needs to change. But what about your less tech-savvy coworkers? Adding a headless CMS to the mix makes collaborating during content creation a breeze for the whole team. And not only that, you gain a proper workflow, role permissions, and all the benefits of structured content that a simple text file just can't offer.

To make sure you're completely of the hook from updating the content on your site, you can use webhooks to help you set up a tool that will pull content via Kontent APIs and generate a new static site whenever content has changed within the CMS.

React-based static site generators

Leverage GraphQL with Gatsby

Gatsby is a React-based static site generator that uses GraphQL for manipulating data. You can connect Gatsby to your Kontent project in two ways.

1. The Gatsby source plugin for Kontent (available as the `@kentico/gatsby-source-kontent` npm package) retrieves data from the Kontent Delivery REST API to build static sites using Gatsby.
   - There's also the `@kentico/gatsby-kontent-components` npm package that contains React components useful when processing Kontent data.
2. The native Delivery GraphQL API lets you query data from your Kontent project. Connect it to your Gatsby app using a Gatsby source plugin.

To find out more details and learn about best practices of working with Gatsby, have a look at

- Gatsby guide – a guide on how to quickly source content for your Gatsby site,
- Gatsby showcases – examples how to implement navigation and how to resolve inline images, components, and content items inside a rich text element.

✓ Using Web Spotlight?

Learn how to implement Web Spotlight using Gatsby Starter to create blazing fast websites.

Create static and dynamic Jamstack sites with Next.js

Next.js is an open-source React framework you can use to achieve consistency by building a static site for visitors and a server-rendered preview for content creators. Here are some resources to help you get started quickly:

- An interactive tutorial to build a Next.js app.
- The official showcase of a Next.js site that uses Kontent as a data source.
- A guide to image optimization with the Next image component.

Vue-based static site generators

The Kontent Nuxt.js module is a module for using the Kontent Delivery JavaScript SDK to create static generated Vue.js applications with Nuxt.js. Nuxt.js is a progressive framework built with Vue.js that lets you build production-ready web apps.

The Kontent source plugin for Gridsome allows you to work with Kontent as your content source to build static sites using Gridsome. Gridsome is a Vue-powered static site generator for building blazing fast static websites. It is data-driven meaning it uses a GraphQL layer to get data from different sources in order to dynamically generate pages from it.

What's next?

- See available JavaScript-based SDKs to develop Kontent powered apps.
- See how to make your static site dynamic by providing dynamic functionality delivered via serverless functions.
- Get your content model ready for Jamstack by following three basic principles.
- Check the Gatsby starters library with templates based on Kontent.
- Learn how to personalize static sites using Kontent and Pardot.