

7 Reasons NOT to Use a Static Site Generator

Trending posts on SitePoint today:

[7 Ways to Make WordPress Simpler for Users](https://www.sitepoint.com/make-wordpress-simpler-users/)

(<https://www.sitepoint.com/make-wordpress-simpler-users/>).

[I Need a Website. What Do I Need to Know About Hosting?](https://www.sitepoint.com/i-need-a-website-what-do-i-need-to-know-about-hosting/)

(<https://www.sitepoint.com/i-need-a-website-what-do-i-need-to-know-about-hosting/>).

[WordPress Version Control with Git](https://www.sitepoint.com/wordpress-version-control-git/) (<https://www.sitepoint.com/wordpress-version-control-git/>).

[10 Easy WordPress Security Tips](https://www.sitepoint.com/10-easy-wordpress-security-tips/) (<https://www.sitepoint.com/10-easy-wordpress-security-tips/>).

[What Is the Right Image Format for Your Website?](https://www.sitepoint.com/what-is-the-right-image-format-for-your-website/)

(<https://www.sitepoint.com/what-is-the-right-image-format-for-your-website/>).

In my [previous article](http://www.sitepoint.com/7-reasons-use-static-site-generator/) (<http://www.sitepoint.com/7-reasons-use-static-site-generator/>), we discussed reasons why you could benefit from using a static site generator. To recap:

A **static site** is a collection of pages contained in basic HTML files. You could hand-write these in a text editor, but managing assets and repeated elements such as navigation can become problematic.

A **content management system** (CMS) stores page content in a database and provides facilities to edit and apply themes. Management becomes easier at the expense of flexibility, performance, server requirements, security and backups.

A **static site generator** (SSG) is a compromise between using a hand-coded static site and a full CMS. You generate an HTML-only website using raw data (such as Markdown files) and templates. The resulting build is transferred to your live server.

STATIC SITE GENERATORS

Popular static site generators include [Jekyll](http://jekyllrb.com/) (<http://jekyllrb.com/>), [Pelican](http://blog.getpelican.com/) (<http://blog.getpelican.com/>), [Hugo](http://gohugo.io/) (<http://gohugo.io/>), and [Metalsmith](http://www.metalsmith.io/) (<http://www.metalsmith.io/>) – see [StaticGen](http://www.staticgen.com/) (<http://www.staticgen.com/>) for many more options.

SSGs appear to offer the benefits of both CMS and static worlds, but they will not be suitable for every project ...

1. You're On Your Own

You won't get far using a static site generator without some development expertise. The process is more difficult than using a CMS, there are fewer reference resources, and you could struggle to find pre-built plugins and templates.

Contrast that with WordPress. A non-technical user may require installation assistance but, once done, they can edit a site and install one of the many thousands of themes and plugins available. They may not have the best bespoke website, but they're running with minimal intervention.

2. Choice Paralysis

There are many [static site generators](http://www.staticgen.com/) (<http://www.staticgen.com/>), but even the most popular tools are used by a tiny proportion of the web community. You'll need time to research, investigate and evaluate the options. The undisputed champion is [Jekyll](http://jekyllrb.com/) (<http://jekyllrb.com/>), but, while you don't necessarily require Ruby expertise, it will help if you've used the language before.

There are many CMSs, but there's one obvious choice: *WordPress*. It powers more than one in every four websites, so help is abundant. Again, it will help if you have some PHP experience, but even a non-developer can create a reasonable website using off-the-shelf themes and plugins.

3. The Initial Setup Time (<https://www.sitepoint.com/>).

Creating your first static site will take time. You'll need to learn the build process, and much of the template code will need to be developed. Deployment scripts may also be necessary.

Developing a custom CMS theme can also be complicated, but pre-built templates are available and assistance is easier to find. File deployment is rarely required following the initial installation.

4. No Administration Interface

Clients may be cautious when faced with a CMS interface. Asking them to create and edit a set of Markdown files may terrify many. You can make the process easier. For example:

- 1 allow them to use their existing CMS, or
- 2 provide simpler workflows such as editing Dropbox files in StackEdit (<https://stackedit.io/>).

But this will further impact your initial development time.

5. Website Consistency

Static sites are flexible: anything contained within source Markdown files can be rendered as page content. Users may be able to include scripts, widgets or numerous undesired items.

A CMS constrains the user, because content is normally bound to a database with specific fields. Control panels prompt the user so it's obvious they must enter a title, body content, excerpt etc. Even if the user enters something in an unexpected field, it won't appear on the website unless it's implemented within the theme template.

6. Managing Large Sites

Consider a website with thousands of pages, regular publications, real-time breaking news and dozens of authors in multiple locations. SitePoint is a great example. It would be possible to manage content using a static site generator, but:

content editing and publishing is more awkward. Editors may require access to the Git repo or shared folders rather than a simple web or app interface.
real-time updates would be delayed because the site must be rebuilt, tested and deployed
build times could increase rapidly and deployment would become cumbersome.

Static site generators are perhaps best suited to sites containing no more than a few hundred pages with a couple of new posts every week. Automated build and deploy processes will be required, and you may reach a point where a CMS becomes a more practical option.

A true static site cannot [\(https://www.sitepoint.com/\)](https://www.sitepoint.com/) as user logins, form filling, search or discussion forums. There are some client-side options such as [lunr.js search \(http://lunrjs.com/\)](http://lunrjs.com/) and [Disqus commenting \(https://publishers.disqus.com/\)](https://publishers.disqus.com/), but your choices will be more limited.

There are solutions such as using a build step which generates a static site containing necessary `<?php ... ?>` code blocks. This should pre-render HTML where possible, e.g. navigation, includes, partials, etc. to ensure pages are more efficient and require less effort from the PHP processor. Unfortunately, you're increasing the build complexity, and it may be of little benefit to some frameworks.

Is a Static Site Right For You?

Before making any decision, examine:

- your project requirements, size, complexity, etc.
- your userbase
- your team's development skills, and
- the hosting and/or deployment factors.

The vast majority of websites rarely exceed a few dozen pages, receive infrequent updates and depend on a developer to make those changes. A CMS is often overkill, and static site generation could ease development and reduce costs. However, persuading your client to abandon their CMS administration panels could be a tougher task ...

For a practical demonstration of how to build a site with a static site generator, see my [How to Create a Static Site with Metalsmith \(http://www.sitepoint.com/create-static-site-metalsmith/\)](http://www.sitepoint.com/create-static-site-metalsmith/) article.



Meet the author

(ht [Craig Buckler \(https://www.sitepoint.com/author/craig-buckler/\)](https://www.sitepoint.com/author/craig-buckler/) 
n" [. \(https://twitter.com/craigbuckler\)](https://twitter.com/craigbuckler) [f \(http://www.facebook.com/craigbuckler\)](http://www.facebook.com/craigbuckler) [in \(http://www.linkedin.com/in/craigbuckler\)](http://www.linkedin.com/in/craigbuckler) [G \(https://github.com/craigbuckler\)](https://github.com/craigbuckler)

Craig is a freelance UK web consultant who built his first page for IE2.0 in 1995. Since that time he's been advocating standards, accessibility, and best-practice HTML5 techniques. He's created enterprise specifications, websites and online applications for companies and organisations including the UK Parliament, the European Parliament, the Department of Energy & Climate Change, Microsoft, and more. He's written more than 1,000 articles for SitePoint and you can find him [@craigbuckler \(http://twitter.com/craigbuckler\)](http://twitter.com/craigbuckler).

