

3 incorrect and 1 correct way to make a blog

Python homework help

Series "Development of static sites with Middleman"

3 incorrect and 1 correct way to make a blog

Maximum fast and full-fledged blog on Middleman and AWS

Continuous integration, Jenkins and Middleman

3 incorrect and 1 correct way to make a blog

AUTHOR:

Kirill Shirinkin

Devops and Infrastructure Consultant, AWS expert, programmer, author of three books, mentor and founder MKDev.me. I design and develop software products. I also write about this and teach people.

My personal site, FODOJ.COM, where you can find everything I think about personal productivity, travel and various books, for about 7 years. It was one of the first sites that I created being a fifteen-year-old schoolboy. Of course, at that time the project looked quite differently. His design and content I will leave behind the scenes. First, because then I will have to blush (and far from design). Secondly, because I first want to tell about the technical side of the project, which changed at least 4 times.

Spoiler: If you still want me to shine for my previous articles, you will drive FODOJ.com to <http://web.archive.org/>. Or google "fashionj.wordpress.com".

V0: only static

Initially, the site lived at Gornozavodsk.com. It was an impressive messenger of JavaScript-whistle glands, half-tissue girls from anime and lines of text that only a schoolboy could write to study web development. V1: Blogger.com and WordPress.com

At some point I decided that I want to blog and started such on Blogger.com. Apparently, that service it seemed uncomfortable to me and I switched to Wordpress.com. It was the last proper technical solution that I accepted on FODOJ.com for the next 7 years. Blog on WordPress.com successfully existed two more than a year, after which I wanted not just some kind of blog, but a full-fledged personal site with steep pieces and incredible flexibility. V2: Welcome to the world of CMS

At that time I had already understood that writing my blog for the blog was a stupid solution and you need to choose from existing options. In this case, I needed flexibility, allowing you to go beyond any time "just a blog" and add, for example, I read the cry at the moment (do not ask).

But if you still wonder what book I read at the moment, then welcome to Goodreads.

From the whole zoo of the engine, I eventually stopped at ExpressionEngine. This is a very cool engine, the first version of which was distributed free of charge. Now the first version is unlikely to be found, and the second one is distributed only for money.

EXPRESSIONENGINE I really liked and served me faith and truth and two years in a row. By the way, if you are looking for something similar to ExpressionEngine, but free, then look at MODX. I used it for some kind of one project many years ago and I liked it. Perhaps it is still good. V3: moment when everything went wrong

In 2012, I was tired of using ExpressionEngine. By the time it became clear that no one was going to produce new chips to the free version. I'm tired of the confusing interface, and I wanted a little more than Fich than the system could offer. Moreover, I decided to completely remake the design. In short, I decided to write my engine for my blog. And wrote. As a result, Fodoj.com twitched two years on a small Rails application.

In fact, this approach has certain advantages. James Edward Gray, famous in the Ruby Community, for example, for his blog also wrote his engine. You can roll out your cunning comment system and imperceptibly to introduce into a blog, for example, an online store and hosting Docker containers (newcomers from Web Virgo and Hipster: It was a joke).

In my case, I needed such "cool" chips like:

Administ for random phrases that are displayed in the header of the site

Markdown Editor for Text

...

Plushka "This site works on my own engine"

Over time, the importance of the first and last items slightly decreased (to zero), and Markdown can be obtained in other places.

The shortcomings of the support of their engine are a little more weight: the time costs of developing unnecessary new features

Okay: it's all the same Ruby on Rails and it is not difficult for it to cry out any needed functionality for

a relaxed evening, full of tea Mate and Orchestra Glen Miller. However, this is a whole evening that I could spend, for example, writing new features for MKDEV or new articles (or something more landed, like a campaign to the movies).

But more importantly: the site does not need new functionality. Over the past year, I added the following killer features: Robots.txt, Sitemap and RSS feed. We have on MKDEV there is even an article about how to do it on Ruby on Rails: the right blog on Rails: Sitemap, Robots and RSS.

In addition, even these trivial things may be many small nuances that developers of existing systems took care of. Speed

There is a certain (and very obvious) speed gain when switching to the Rails of the application spinning on a free Heroku, to a static site located on AWS S3 and spelling through CloudFront. Support

New Vulnerability in Rails? Urgently update Fjengine! New vulnerability in Ruby? Urgently update Fjengine! Heroku changed routing rules? Well, you understood. Do you know how many different vulnerabilities in a simple HTML file? That's exactly what. V4: Light Future with Middleman

Meet Middleman: Written on Ruby Static Site Generator. The principle of operation is simple: you create the site using all comfortable things accelerating the development. And then generate one command the final set of HTML \ CSS, which and pour on hosting.

From the Middleman box supports all the joys of modern web development like SCSS, ERB templates, partials, localization. It is also written for it a ton of extensions for any needs, including a debt on S3 and highlighting pieces of code.

There are several popular alternatives for Middleman. The most significant is Octopress and Jekyll. The latter is even used for GitHub Pages. I chose Middleman because Intrudka, for him much more ready plugins, it is easier to expand and I worked with him in the past. The official plugin for the blog supports everything I need. I do not exclude that all that we will do in the following articles of the series can be achieved using Octopress and Jekyll. I have no experience with any other, so I will be glad to hear in the comments of the success of their use. What's next?

In the following articles of the series, I will describe the entire migration process with FjEngine in detail on Middleman and will tell you how to automate the deploy of Middleman sites.

We tell how to become a better developer, how to maintain and effectively apply your skills. Information about jobs and promotions exclusively for more than 8,000 subscribers. Join!

Our newsletter: