

STATIC SITE GENERATION FOR THE MASSES

Denis Defreyne
FOSDEM 2013

Me

Me

- ▶ Denis Defreyne

Me

- ▶ Denis Defreyne
twitter.com/ddfreyne

Me

- ▶ **Denis Defreyne**
twitter.com/ddfreyne
github.com/ddfreyne

Me

- ▶ **Denis Defreyne**
twitter.com/ddfreyne
github.com/ddfreyne
- ▶ **Software Engineer**

Me

- ▶ **Denis Defreyne**
twitter.com/ddfreyne
github.com/ddfreyne
- ▶ **Software Engineer**
- ▶ **Author of nanoc**

What exactly does a
STATIC SITE GENERATOR
do?



Mark-
down

Markdown

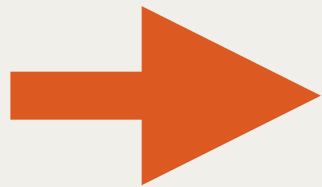
In `*exactly*` one month, there's FOSDEM again! I can barely control my excitement!

`* First element of a list`

`* Second element`



Mark-
down



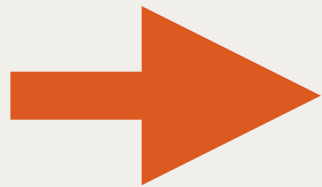
Mark-
down

HTML
fragment

HTML from Markdown

```
<p>In exactly one month, there's  
FOSDEM again! I can barely control my  
excitement!</p>
```

```
<ul>  
  <li>First element of a list</li>  
  <li>Second element</li>  
</ul>
```



Mark-
down

HTML
fragment

Layout



Mark-
down

HTML
fragment

Layout

```
<!DOCTYPE html>
<html>
  <head>
    <title><%= item[:title] %></title>
  </head>
  <body>
    <%= yield %>
  </body>
</html>
```

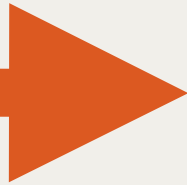
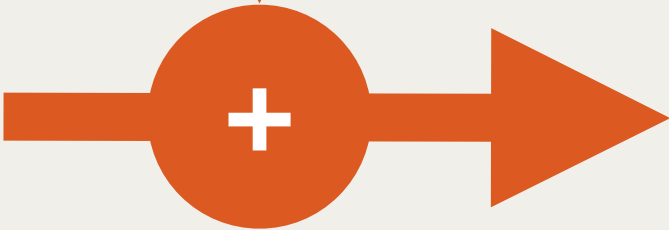
Layout



Mark-down

HTML
fragment

Layout



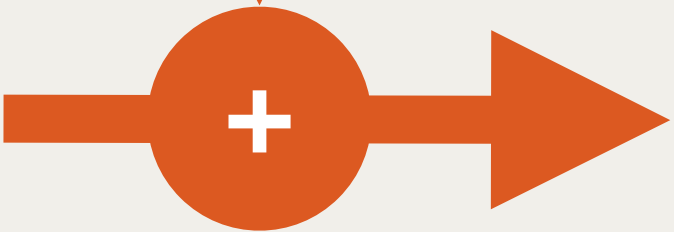
Mark-down

HTML fragment

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>One Month until FOSDEM!</title>
  </head>
  <body>
    <p>In <em>exactly</em> one month, there's FOSDEM
    again! I can barely control my excitement!</p>
    <ul>
      <li>First element of a list</li>
      <li>Second element</li>
    </ul>
  </body>
</html>
```


Layout



Mark-down

HTML fragment

HTML

VPS + 96 MB RAM + Ruby =
TOO SLOW!

nanoc

ナノツク



CMS

Content Management System

~~CMS~~

Content Management System

SSG

Static Site Generator

Road map

Road map

1. What are SSGs?

Road map

1. What are SSGs?
2. What are SSGs capable of?

Road map

1. What are SSGs?
2. What are SSGs capable of?
3. Lessons Learnt & Future Work

What are SSGs?

Tiny example – pages

content/

about.md

blog/

2013-01-02-one-month-until-fosdem.md

2013-02-03-fosdem-was-awesome.md

Tiny example – pages

title: One Month until FOSDEM!

In exactly one month, there's FOSDEM again!
I can barely control my excitement!

Tiny example – layouts

```
<html>
  <head>
    <title><%= item[:title] %></title>
  </head>
  <body>
    <%= yield %>
  </body>
</html>
```


Tiny example – result

```
<html>  
  <head>  
    <title>One Month until FOSDEM!</title>  
  </head>  
  <body>  
    In exactly one month, there's FOSDEM again! I  
    can barely control my excitement!  
  </body>  
</html>
```

Tiny example - result

output/

about/index.html

blog/2013/01/one-month-until-fosdem/index.html

blog/2012/02/fosdem-was-awesome/index.html

nanoc is a *static site generator*, fit for building anything from a small personal blog to a large corporate web site.

The [GitHub API](#) web site is written using nanoc. Several others API documentation projects also use nanoc: take a look at the [App.net API](#), the [Flatlr API](#) and the [Spree API](#), for instance.

Are you interested? [Install nanoc](#) and [follow the tutorial](#)!



The current release is **3.4.3**, released on December 9, 2012. Check out the [release notes](#) for details.



[Install nanoc](#)

A simple `gem install nanoc` will get you going, but we've provided detailed



[Take the tutorial](#)

This ten-minute tutorial will show you how nanoc works, and teach you the basics —



[Learn more](#)

nanoc comes with vast amounts of documentation, covering everything from



The GitHub API

This describes the resources that make up the official GitHub API v3. If you have any problems or requests please contact [support](#).

For the new API v3, start browsing the resources on the right »

View the [API Changelog](#) for information on existing and planned changes to the API.

↔ Current Version

```
Accept: application/vnd.github.beta+json
```

The GitHub API version is currently in beta. The `beta` media type property will be valid until sometime in 2013. A notice will be given closer to the actual date.

We consider the "beta" API unchangeable. [File a support issue](#) if you have problems.

Expected Changes

These changes are *not* implemented, just planned for the next major API version.

- Standardize on existing `*_url` attributes for hypermedia. Remove all `_links` objects.
- The `'/repos/:owner/:repo/hooks/:id/test'` action becomes `'/repos/:owner/:repo/hooks/:id/tests'`.
- The `'/gists/:id/fork'` action becomes `'/gists/:id/forks'`.
- Gist forks/history objects become separate API calls.
- Gist files object is not returned on Gist listings.
- Commit schema will change to be [more consistent](#).
- `master_branch` becomes `default_branch`.
- `integrate_branch` on the [repo API](#) will no longer be returned.
- Use the `private` attribute when creating a private repository, instead of setting `public` to false.
- Use JSON to POST to the `"repos/:owner/:repo/forks"` endpoint, instead of a query string.

| | |
|-----------|---------------|
| ▶ | Overview |
| ▶ | Activity |
| ▶ | Gists |
| ▶ | Git Data |
| ▶ | Issues |
| ▶ | Orgs |
| ▶ | Pull Requests |
| ▶ | Repositories |
| ▶ | Users |
| ▶ | Search |
| Gitignore | |
| Markdown | |

This website is a [public GitHub repository](#). Please help us by forking the project and adding to it.



MYST ONLINE URU — LIVE —

 HOME

 PLAY

 ABOUT

 FORUMS

 DONATE

 DEVELOPERS

LATEST NEWS

MYST ONLINE: URU LIVE AGAIN

Welcome back, again.

The Ages of *Uru* are available again. We've opened all the Ages, and added most of the goodies in *MO:UL*. We're referring to it as *MO:ULagain*—feel free to explore and enjoy.

And the cost is pretty good. **Free!**

To begin playing, click on the [PLAY](#) link to create an account.

If you're so inclined, we've provided a way for you to [DONATE](#) to help offset our costs, and show your support. Five or ten bucks would go a long way toward helping pay the server and people bills.

We have also made most of the sources to *MO:UL* open source. You can see how *MO:UL* ticks, create your own shards, make improvements, fix bugs, create new ages. We will take some of the best improvements, ages and fixes from the fans and apply them to the live *MO:UL* server. If this





GETTING STARTED

- [Responses](#)
- [Migrations](#)
- [Rate Limits](#)
- [Data Formats](#)
- [Messaging](#)

AUTHENTICATION

- [Web Flows](#)
- [Password Flow](#)
- [App Access Token Flow](#)
- [Identity Delegation](#)

RESOURCES

- [User](#)
 - [Lookup](#)
 - [Profile](#)
 - [Following](#)
 - [Muting](#)
 - [Post Interactions](#)
- [Post](#)
 - [Lookup](#)
 - [Lifecycle](#)
 - [Streams](#)
 - [Replies](#)
 - [Reposts](#)
 - [Stars](#)
- [Channel](#)
 - [Lookup](#)
 - [Lifecycle](#)
 - [Subscriptions](#)
- [Message](#)
 - [Lookup](#)
 - [Lifecycle](#)
- [Stream](#)
 - [Lifecycle](#)
- [Filter](#)

App.net API Documentation

App.net is a subscription-based, ad-free social network and API, a home for meaningful interactions and a platform developer ecosystem. We rely on our developers and our developers can rely on us for a robust, responsive and transparent platform.

Welcome aboard!

This page is your starting point to getting familiar with the App.net developer platform. Our aim is to get you up and running as quickly and easily as possible. What you see here is always under revision and we always want you to give us your feedback.

Tracking news and announcements

To see the latest updates to the API, follow [@adnapi](#). You may also want to subscribe to the [App.net blog](#) for general news and the [App.net developer blog](#) for developer related news.

- (12/13/2012) [Introducing the Messaging API](#) — check out the documentation for [Channels](#) and [Messages](#) and the [Messaging overview](#).
- (12/13/2012) [Documentation has a new look and organization!](#)

Getting help and providing feedback

There are numerous ways for developers to get help utilizing the platform and to provide feedback. During typical San Francisco business hours there are App.net staff members available to answer questions in our [developer chatroom](#). For general account/service related issues email support@app.net. For developer inquiries about the API, our Terms of Service, letting us know about something you're working on, etc., email developers@app.net. In addition to the above you can report bugs and provide feedback on the API using the [issue tracker](#) or by using pull requests and commit comments on the [Github repo for the API](#) or the [API docs](#). You can also contact our Developer Advocate [@orian](#).

Terms of Service

All usage of the App.net API is governed by our [Developer Terms of Service](#). We spent a great deal of time making this document readable, concise, and developer friendly. Please take the time to review it.

Registering your app

Establishing a robust ecosystem of great apps is critical for our success. We're always looking for more ways to promote



REST API

[Introduction](#)

[Authorization](#)

[Resources](#)

[Flattr](#)

[Things](#)

[Subscriptions](#)

[Users](#)

[Activities](#)

[Categories](#)

[Languages](#)

[Console](#)

[Changelog](#)

[Clients](#)

[Policy](#)

[Embedded buttons](#)

[Auto-submit URL](#)

[Flattr in feeds and HTML](#)

[Partner site integration](#)

[Tools](#)

[Questions](#)

Welcome

We offer several ways to communicate and integrate with different parts of [Flattr](#).

REST API

Use our [REST API](#) if you want to integrate with Flattr. It offers OAuth2 authentication of Flattr users, ways to create, delete and update things and much more.

Button

If you want to include Flattr buttons on your page check out the [embedded button](#) or the [auto-submit URL](#) documentations.

The documentation

This documentation site is available on [GitHub](#).



Visit Flattr.com to learn more about Flattr.

UGLY BETTY

WEEKENDS JUST GOT BETTYER.



HOME

CHARACTERS

VIDEOS

PHOTOS

LOCAL LISTINGS



FIND STATIONS & AIR TIMES IN YOUR AREA

Enter Your Zip Code

GO!

MEET THE CHARACTERS

BETTY SUAREZ



WILHELMINA SLATER

DANIEL MEADE



AMANDA TANEN

[More »](#)

PHOTO GALLERY



VIDEOS



"There's been a mistake."
Betty gets brushed off before her interview at Meade Publications even begins.
[Watch »](#)

Click here to Bettyize yourself with the

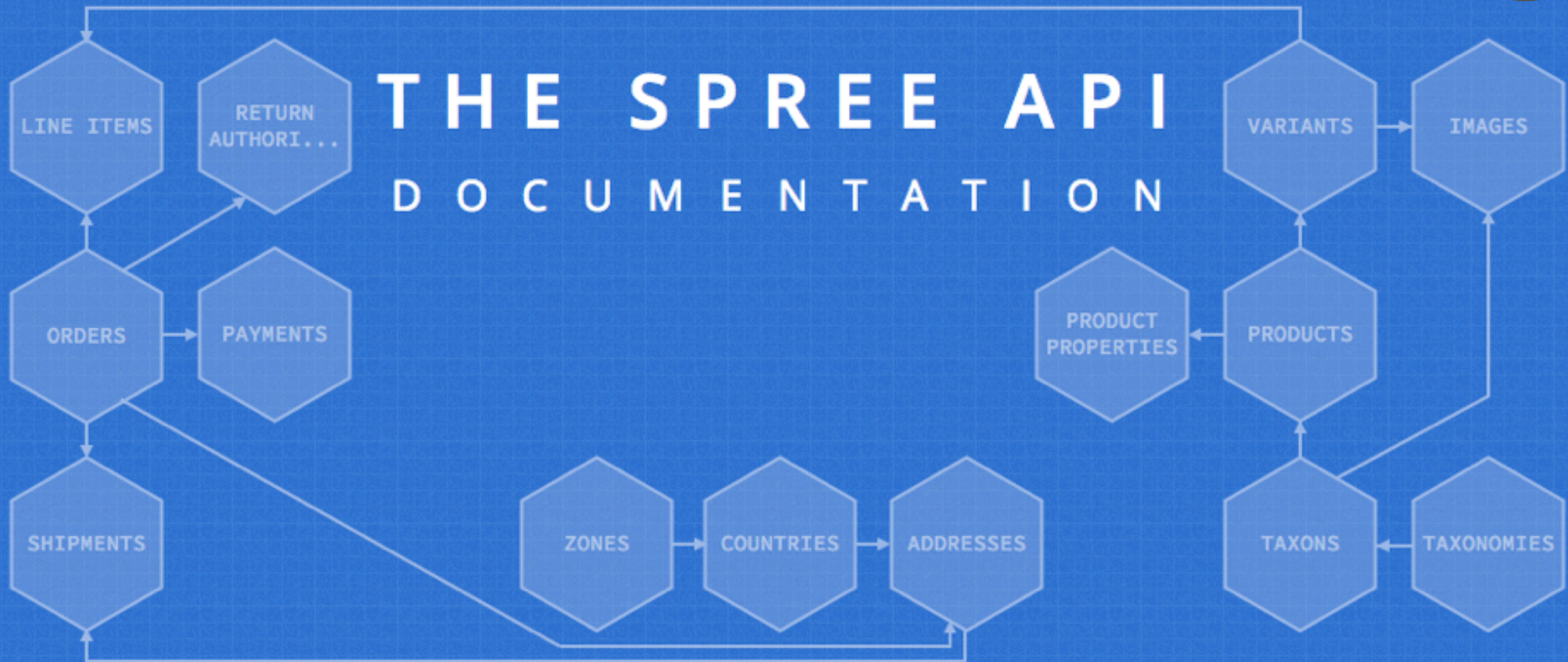
BE BETTYER

Photo Booth



and then share the





[Get Started](#)

The Spree API

This site covers the inner working of Spree's RESTful API. It assumes a basic understanding of the principles of REST.

The REST API is designed to give developers a convenient way to access data contained within Spree. With a standard read/write interface to store data, it is now very simple to write third party applications

Latest Changes

- [api] Added support for updating...
2013-01-17T17:28:13Z @LBRapid
- [api] Added a CheckoutsControlle...
2013-01-16T20:56:19Z @LBRapid



beer
open source
lightning talks



devrooms
5000+ hackers
456 lectures

Brussels / 2 & 3 February 2013

[schedule](#)

FOSDEM is a free event that offers open source communities a place to meet, share ideas and collaborate.

It is renowned for being highly developer-oriented and brings together 5000+ geeks from all over the world.

No registration necessary.

SPONSORS



NEWS

2013-01-16

Second batch of speaker interviews

We have just published the

2013-01-14

Call for volunteers

FOSDEM needs you! Every year, the FOSDEM team is

2013-01-11

Accepted Stands

The stands are spread over two buildings, the K building

2013-01-09

First FOSDEM 2013 speaker interviews

Just like previous editions we

Ace - awestruct - Blacksmith - Blogofile - Bonsai - Cactus
- Chisel - coleslaw - Composer - cyrax - Deplot - Fairytale
- FMPP - Frank - fugitive - Graze - Hakyll - Hammer -
Hastie - Hobix - Hyde - Jekyll - jinjet - jkl - Korma -
Machined - Magneto - Markbox - Markdoc - Middleman -
mynt - **nanoc** - Nikola - Octopress - Pagegen - Pelican -
Petrify - Phrozn - PieCrust - Pilcrow - Pith - poole - Punch
- Quill - RakeWeb - Rassmalog - Rizzo - Rog - romulus -
Rote - RubyFrontier - Ruroh - Second Crack - Socrates -
Speechhub - Stasis - staticjinja - StaticMatic -
StaticMatic2 - StaticWeb - Tahchee - Ultra simple Site
Maker - Webby - webgen - Website Meta Language -
Wintersmith - wok - yassg - yst - Yurt CMS - ZenWeb

Ace - awestruct - Blacksmith - Blogofile - Bonsai - Cactus
- Chisel - coleslaw - Composer - cyrax - Deplot - Fairytale
- FMPP - Frank - fugitive - Graze - **Hakyll** - Hammer -
Hastie - Hobix - Hyde - **Jekyll** - jinjet - jkl - Korma -
Machined - Magneto - Markbox - Markdoc - **Middleman** -
mynt - **nanoc** - Nikola - **Octopress** - Pagegen - Pelican -
Petrify - Phrozn - PieCrust - Pilcrow - Pith - poole - Punch
- Quill - RakeWeb - Rassmalog - Rizzo - Rog - romulus -
Rote - RubyFrontier - Ruroh - Second Crack - Socrates -
Speechhub - Stasis - staticjinja - StaticMatic -
StaticMatic2 - StaticWeb - Tahchee - Ultra simple Site
Maker - Webby - webgen - Website Meta Language -
Wintersmith - wok - yassg - yst - Yurt CMS - ZenWeb

Ace - awestruct - Blacksmith - Blogofile - Bonsai - Cactus
- Chisel - coleslaw - Composer - cyrax - Deplot - Fairytale
- FMPP - Frank - fugitive - Graze - **Hakyll** - Hammer -
Hastie - Hobix - Hyde - **Jekyll** - jinjet - jkl - Karma -
Machined - Magneto - Markbox - **Melican** -
mynt - **nanoc** - Nikola - **Orange** - **Relican** -
Petrify - Phrozn - **Pluto** - **Pool** - poole - Punch
- Quill - **Rango** - Rizzo - Rog - romulus -
Rote - **Ruroh** - Second Crack - Socrates -
Speccnub - Stasis - staticjinja - StaticMatic -
StaticMatic2 - StaticWeb - Tahchee - Ultra simple Site
Maker - Webby - webgen - Website Meta Language -
Wintersmith - wok - yassg - yst - Yurt CMS - ZenWeb

INCOMPLETE

HEADSMART

Dynamic header hierarchy

download
.ZIP

download
.TGZ

Welcome to Headsmart

Headsmart is a jQuery plugin which applies header hierarchy dynamically based on which headers are present.

When building templates, we can't put faith in our users that they'll choose the header tag hierarchy that we prefer. While we'd love to assume that they'd know we'd like them to start with an `<h3>`, more likely they'll start with an `<h1>`. This is where Headsmart comes in.

Headsmart looks at the element you supply it, and assigns header classes levels 1 – 6 to the header tags that are actually present. That means that the largest header will still look like your level 1 header whether it's a `<h1>` or `<h3>`.

Usage

1. Since you will be styling your headers with classes rather than their actual elements, you'll want to use some kind of [css reset](#) to get all your headers looking the same.
2. Now that your css is all equal, simply use headsmart on the element containing your content. Headsmart only looks for direct children of the element supplied, so nested headers will not be effected `$('#site-content').headsmart()`
3. Style the classes generated by headsmart as you please: `.header-level-1`

Minimal

A Theme for GitHub Pages

[View the Project on GitHub](#)
orderedlist/minimal

Download
ZIP File

Download
TAR Ball

Fork On
GitHub

This project is maintained by [Steve Smith](#)

Hosted on GitHub Pages — Theme by [orderedlist](#)

Let's Test Some Markup

View the [source of this content](#).

Let's get the whole "linebreak" thing out of the way. The next paragraph contains two phrases separated by a single newline character:

Roses are red
Violets are blue

The next paragraph has the same phrases, but now they are separated by two spaces and a newline character:

Roses are *emphasized*
Violets are **strong**

A bit of the GitHub spice

In addition to the changes in the previous section, certain references are auto-linked:

- SHA: [be6a8cc](#)
- User@SHA ref: [mojombo@be6a8cc](#)
- User/Project@SHA: [mojombo/god@be6a8cc](#)
- #Num: [#1](#)
- User/#Num: [mojombo#1](#)
- User/Project#Num: [mojombo/god#1](#)

These are dangerous goodies though, and we need to make sure email addresses don't get mangled:

My email addy is tom@github.com.

Math is hard, let's go shopping

In first grade I learned that $5 > 3$ and $2 < 7$. Maybe some arrows. $1 \rightarrow 2 \rightarrow 3$. $9 \leftarrow 8$

OCTOPRESS

A blogging framework for hackers.



Blog

Documentation

Help

Source

Search



JUL 23RD, 2011 • ∞

Octopress 2.0 Surfaces

Octopress is a framework designed by [Brandon Mathis](#) for [Jekyll](#), the blog aware static site generator powering [Github Pages](#). To start blogging with Jekyll, you have to write your own HTML templates, CSS, Javascripts and set up your configuration. But with Octopress All of that is already taken care of. Simply [clone or fork Octopress](#), install dependencies and the theme, and you're set.

What's new in 2.0?

Short answer: Everything. Octopress is now based on [mojombo/jekyll](#) has been completely rewritten from the ground up with a mountain of goodies.

Latest Tweets

I'm working on paginated posts, but I want to point read-later services to a unified page. 3w

Ideas? github.com/imathis/octopr...

/cc [@marcoarment](#)

 Follow [@octopress](#)

Why static sites?

Why static sites?

1. Fast

Why static sites?

1. Fast
2. Secure

Why static sites?

1. Fast
2. Secure
3. Works with your own tools (vim!)

Why static sites?

1. Fast
2. Secure
3. Works with your own tools (vim!)
4. Content is in plain text

Why static sites?

1. Fast
2. Secure
3. Works with your own tools (vim!)
4. Content is in plain text
5. Easy to deploy

Why static sites?

1. Fast
2. Secure
3. Works with your own tools (vim!)
4. Content is in plain text
5. Easy to deploy
6. Easy to preview

Why static sites?

1. Fast
2. Secure
3. Works with your own tools (vim!)
4. Content is in plain text
5. Easy to deploy
6. Easy to preview
7. Makes collaboration easy



Code Network Pull Requests 0 Graphs

GitHub API documentation — [Read more](#)
<http://developer.github.com>



Clone in Mac
 ZIP
 HTTP
 SSH
 Git Read-Only
 git://github.com/github/developer.github.com.␣
 Read-Only access

branch: master
 Files
 Commits
 Branches 16
 Tags

Switch branches/tags

Find or create a branch...

Branches Tags

- ✓ announce-org-repo-pagination
- changes-links
- clarify-state
- clearer-owner-repo
- commit-status-examples
- gh-pages
- gist-comment-uris
- introduce-v3-media-type
- ✓ master
- org-members-redirection
- org-repos-sources-and-forks

804 commits

latest commit 50e82d1885

| | |
|---|--|
| ... | |
| Note that state is required only if submitted in step 1 [bhuga] | |
| Add status to header [pengwynn] | |
| Make example issue number and URLs agree [pengwynn] | |
| Add a script/bootstrap [tclm] | |
| Add status text [pengwynn] | |
| Move static resources around, adding new datasource to handle them [tclm] | |
| Replace .DS_Store.bundle with .DS_Store [leereilly] | |
| Ease up, rbenv [pengwynn] | |
| Make rvm use 1.9.3. [Mab879] | |
| Bump nanoc for sass compat [pengwynn] | |
| Bump nanoc for sass compat [pengwynn] | |
| Updated README.md [parndt] | |

Rakefile a month ago add a ' to message prompt [rick]

What are SSGs capable of?

Arbitrary metadata

Arbitrary metadata

title: Braid
kind: review
subtype: game
year: 2008
rating: 5

Arbitrary metadata

```
<ul>
  <% @item.children do |review| %>
    <li>
      <%= '★' * review[:rating] %>
      <%= review[:title] %>
    </li>
  <% end %>
</ul>
```

Arbitrary metadata

```
<ul>  
  <li>★★★★☆ Borderlands</li>  
  <li>★★★★★ Braid</li>  
  <li>★★★★☆ Legend of Grimrock</li>  
  <li>★★★★★ Max Payne</li>  
  <li>★★★★☆ Minecraft</li>  
</ul>
```

Filters

Filters – Markdown

```
A First Level Header
```

```
=====
```

```
A Second Level Header
```

```
-----
```

```
Now is the time for all good men to come to the aid of their  
country. This is just a regular paragraph.
```

- ```
* Quick brown fox
```
- ```
* Lazy dog
```

Filters – Textile

```
h1. Give RedCloth a try!
```

```
A *simple* paragraph with  
a line break, some emphasis and a "link":http://redcloth.org
```

```
* an item  
* and another
```

```
# one  
# two
```

Filters – ERB

```
<!DOCTYPE html>
<html>
  <head>
    <title><%= item[:title] %></title>
  </head>
  <body>
    <%= yield %>
  </body>
</html>
```

Filters – Mustache

```
<!DOCTYPE html>
<html>
  <head>
    <title>{{ title }}</title>
  </head>
  <body>
    {{ yield }}
  </body>
</html>
```

Filters – Haml

```
!!!  
%html  
  %head  
    %title= @item[:title]  
  %body  
    = yield
```

Filters – Colorize Syntax

```
<pre><code class="language-ruby">  
def my_function  
  (1..10).map do |i|  
    "#{i} is #{i % 2 == 1 ? 'odd' : 'even'}"  
  end  
end  
</code></pre>
```

Filters – Colorize Syntax

```
def my_function
  (1..10).map do |i|
    "#{i} is #{i % 2 == 1 ? 'odd' : 'even'}"
  end
end
```

Filters – Typograby

"Typograby makes HTML look smarter & better, don't you think?"

“Typograby makes HTML look smarter & better, don't you think?”

Filters – Custom

```
class CensorFilter < Nanoc::Filter

  identifier :censor

  def run(content, params={})
    content.gsub('nanoc sucks', 'nanoc rocks')
  end

end
```

Rules

Rules

```
compile '*' do
  if @item[:kind] == 'article'
    filter :bluecloth
  end

  layout 'default'
end
```

Rules

```
compile '/images/*/' do  
  filter :img_optimize  
end
```

Rules

```
compile '/images/*/' do  
  filter :img_optimize  
end
```

```
compile '/images/*/', :rep => :thumbnail do  
  filter :scale, :width => 140  
end
```

Rules

```
compile '/blog/*/',' :rep => :mp3 do  
  filter :say  
  filter :lame  
end
```

Rules versus Makefiles

Rules versus Makefiles

- ▶ Push vs. pull

Rules versus Makefiles

- ▶ Push vs. pull
- ▶ Cleaner syntax

Rules versus Makefiles

- ▶ Push vs. pull
- ▶ Cleaner syntax
- ▶ Automatic dependency tracking

Helpers

```
module Nanoc::Helpers::Tagging

  # Find all items with the given tag.
  #
  # @param [String] tag The tag for which to find all items
  #
  # @return [Array] All items with the given tag
  def items_with_tag(tag)
    @items.select do |i|
      (i[:tags] || []).include?(tag)
    end
  end
end

end
```

Helpers – example

```
<h1>Tags for <%= @item[:tag] %></h1>
```

```
<ul>
```

```
<% items_with_tag(@item[:tag]).each do |i| %>
```

```
  <li><%= link_to(@item[:tag], i) %></li>
```

```
<% end %>
```

```
</ul>
```

Workflow

Workflow

1. Development

Workflow

1. Development

1. Version controlled

Workflow

1. Development

1. Version controlled
2. Branched development

Workflow

1. Development

1. Version controlled
2. Branched development
3. Pull requests

Workflow

1. Development

1. Version controlled
2. Branched development
3. Pull requests

2. Release

Workflow

1. Development

1. Version controlled
2. Branched development
3. Pull requests

2. Release

1. Preview

Workflow

1. Development

1. Version controlled
2. Branched development
3. Pull requests

2. Release

1. Preview
2. Check

Workflow

1. Development

1. Version controlled
2. Branched development
3. Pull requests

2. Release

1. Preview
2. Check
3. Deploy

Data Sources

Data Sources – nanoc

nanoc site



FS

YARD

CLI

nanoc CLI – definition

```
name      'dostuff'  
usage     'dostuff [options]'  
aliases   :ds, :stuff  
summary   'does stuff'  
description 'This command does a lot!'
```

[GITHUB.COM/DDFREYNE/CRI](https://github.com/DDFREYNE/CRI)

nanoc CLI – generated help

NAME

`dostuff` - does stuff

USAGE

`dostuff` [options]

DESCRIPTION

This command does a lot of stuff. I really mean a lot.

OPTIONS

| | |
|-------------------------|----------------------------|
| <code>-h --help</code> | show help for this command |
| <code>-m --more</code> | do even more stuff |
| <code>-s --stuff</code> | specify stuff to do |

[GITHUB.COM/DDFREYNE/CLI](https://github.com/DDFREYNE/CLI)

check Command

Summary

```
check [options] [names]
  run issue checks
```

Description

Run issue checks on the current site. If the `--all` option is passed, all available issue checks will be run. If the `--deploy` option is passed, the issue checks marked for deployment will be run.

Options

- `-a --all`
run all checks
- `-l --list`
list all checks
- `-d --deploy`
run checks for deployment

Command details

[↑ Back to Commands](#)

Name

check

This site is [hosted on GitHub](#). If you want to help out, do report issues and submit pull requests!

nanoc filter

```
class CensorFilter < Nanoc::Filter

  identifier :censor

  # Some documentation goes here..
  def run(content, params={})
    content.gsub('nanoc sucks', 'nanoc rocks')
  end

end
```

Class: Nanoc::Filters::ColorizeSyntax

Inherits: [Nanoc::Filter](#)[show all](#)Defined in: `lib/nanoc/filters/colorize_syntax.rb`

Constant Summary

DEFAULT_COLORIZER =

The default colorizer to use for a language if the colorizer for that language is not overridden.

`:coderay`

SIMON_HIGHLIGHT_OPT_MAP =

```
{
  :wrap => '-W',
  :include_style => '-I',
  :line_numbers => '-l',
}
```

Constants inherited from Nanoc::Filter`Nanoc::Filter::TMP_BINARY_ITEMS_DIR`

Instance Attribute Summary

Attributes inherited from Nanoc::Filter`#assigns`

Instance Method Summary

[\(collapse\)](#)

- (String) `coderay`(code, language, params = {}) private
Runs the code through `CodeRay`.
- (String) `dummy`(code, language, params = {})
Returns the input itself, not performing any code highlighting.
- (String) `pygmentize`(code, language, params = {}) private
Runs the content through `pygmentize`, the commandline frontend for `Pygments`.
- (String) `pygmentsrb`(code, language, params = {}) private
Runs the content through `Pygments` via `pygments.rb`.
- (String) `run`(content, params = {})
Syntax-highlights code blocks in the given content.

`(String) simon_highlight(code, language, params = {})` private

Invoking the filter with custom parameters

```
filter :colorize_syntax,  
      :colorizers => { :ruby => :coderay },  
      :coderay     => { :line_numbers => :list }
```

Parameters:

- **content** (*String*) — The content to filter
- **params** (*Hash*) (*defaults to: {}*) — a customizable set of options

Options Hash (params):

- **:default_colorizer** (*Symbol*) — default: `DEFAULT_COLORIZER` — The default colorizer, i.e. the colorizer that will be used when the colorizer is not overridden for a specific language.
- **:syntax** (*Symbol*) — default: `:html` — The syntax to use, which can be `:html`, `:xml` or `:xhtml`, the latter two being the same.
- **:colorizers** (*Hash*) — default: `{}` — A hash containing a mapping of programming languages (symbols, not strings) onto colorizers (symbols).
- **:outside_pre** (*Boolean*) — default: `false` — `true` if the colorizer should be applied on code elements outside `pre` elements, `false` if only code elements inside `pre` elements should be colored.
- **:is_fullpage** (*Symbol*) — default: `false` — Whether to treat the input as a full HTML page or a page fragment. When `true`, HTML boilerplate such as the doctype, `html`, `head` and `body` elements will be added.

Returns:

- (*String*) — The filtered content

[\[View source\]](#)

```
- (String) simon_highlight(code, language, params = {})
```

This method is part of a private API. You should avoid using this method if possible, as it may be removed or be changed in the future.

Runs the content through [Highlight](#).

Parameters:

- **code** (*String*) — The code to colorize
- **language** (*String*) — The language the code is written in
- **params** (*Hash*) (*defaults to: {}*) — a customizable set of options

Options Hash (params):

- **:style** (*String*) — The style to use

Returns:

- (*String*) — The colored output

Since:

- 3.2.0

[\[View source\]](#)

ColorizeSyntax Filter

Summary

Syntax-highlights code blocks in the given content. Code blocks should be enclosed in `pre` elements that contain a `code` element. The code element should have an indication of the language the code is in. There are two possible ways of adding such an indication:

1. A HTML class starting with `language-` and followed by the code language, as specified by HTML5. For example, `<code class="language-ruby">`.
2. A comment on the very first line of the code block in the format `#!language` where `language` is the language the code is in. For example, `#!ruby`.

Options for individual colorizers will be taken from the `{#run}` options' value for the given colorizer. For example, if the filter is invoked with a `:coderay => coderay_options_hash` option, the `coderay_options_hash` hash will be passed to the CodeRay colorizer.

Currently, the following colorizers are supported:

- `:coderay` for [Coderay](#)
- `:pygmentize` for [pygmentize](#), the commandline frontend for [Pygments](#)
- `:pygmentsrb` for [pygments.rb](#), a Ruby interface for [Pygments](#)
- `:simon_highlight` for [Highlight](#)

Additional colorizer implementations are welcome!

For details, check out the [API documentation](#)

Filter details

[↑ Back to Filters](#)

Name

ColorizeSyntax

Identifier

`:colorize_syntax`

API documentation

[Nanoc::Filters::ColorizeSyntax](#)

This site is [hosted on GitHub](#). If you want to help out, do report issues and submit pull requests!

The default colorizer, i.e. the colorizer that will be used when the colorizer is not overridden for a specific language.

:syntax

The syntax to use, which can be `:html`, `:xml` or `:xhtml`, the latter two being the same.

:colorizers

A hash containing a mapping of programming languages (symbols, not strings) onto colorizers (symbols).

:outside_pre

true if the colorizer should be applied on `code` elements outside `pre` elements, false if only `code` elements inside `pre` elements should be colorized.

:is_fullpage

Whether to treat the input as a full HTML page or a page fragment. When true, HTML boilerplate such as the doctype, `html`, `head` and `body` elements will be added.

Examples

```
<pre><code class="language-ruby">
def foo
  "asdf"
end
</code></pre>
```

Using a class to indicate type of code be highlighted

```
<pre><code>
#!ruby
def foo
  "asdf"
end
</code></pre>
```

Tagging Helper

Summary

Provides support for managing tags added to items.

For details, check out the [API documentation](#).

Functions

`tags_for(item, params) → String`

Returns a formatted list of tags for the given item as a string.

`items_with_tag(tag) → Array`

Find all items with the given tag.

`link_for_tag(tag, base_url) → String`

Returns a link to to the specified tag.

Helper details

[↑ Back to Helpers](#)

Name

Tagging

API documentation

[Nanoc::Helpers::Tagging](#)

This site is [hosted on GitHub](#). If you want to help out, do report issues and submit pull requests!

Data Sources – FOSDEM

FOSDEM site



FS

PentaBarf

Web development

[↑ Room: K.1.105](#)
[Calendar: iCal, xCal](#)

| | | | | | | | | | | |
|--------|----|------------|-------------|----|----|----|----|----|----|----|
| | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| Sunday | | Scaling... | Static s... | | | | | | | |

| Event | Speakers | Start | End |
|---------------------------------------|----------------|-------|-------|
| Sunday | | | |
| Scaling PHP with HipHop | Sara Golemon | 10:00 | 10:50 |
| Static site generation for the masses | Denis Defreyne | 11:00 | 11:50 |

| ← / | Janson | K.1.105 | Ferrer | Chavanne | Lameere | H.1301 | H.1302 | H.1308 | H.1309 | H.2213 | H.2214 | AW1.120 | AW1.121 | AW1.125 | AW1.126 | Guillissen |
|-------|------------|--|--|---|--|---|---|--|---|---|---|---|--|---|---|--|
| 09:15 | | | | | | | | | Integrat... Voice through Adhears... | Introduc... LibreOffi... 4.0 | | | | | | |
| 09:30 | | | | State of the OpenSt... Union, 2013 | | | | | | The Upgrade of the LibreOffi... Template Reposit... | Welcome in the MySQL & Friends Devroom 2013 | Introduc... to Ada for Beginning and Experie... Program... | openbar... a possible... solution for ecological regulation | | Meet the smalltal... show us your project | |
| 09:45 | | | | | | | | | SIP is hard, let's go shopping! | How to ensure that eGovern... forms are compati... with Free Software? | 5 Percona Toolkit tools that could save your day | | | | | |
| 10:00 | Samba4 | Scaling PHP with HipHop | Actionaz: Automa... for everyone | OpenSt... 21st Century App Architec... and Cloud Operati... | Making a Hackers... Smart | systemd in Debian | | | | Simple introduc... to LibreOffi... QA | What's new in MySQL | | | | Pandora - a mobile Linux computer including gaming controls | |
| 10:15 | | | | | | | | How GNOME Obsoleted its "Enable Accessi... Setting | Propelling Asterisk into new SIP grounds | General easy hacks for new hackers | Tools and Techniq... for Higher Reliability Software | PostGIS 2.0 and beyond | | | | Étoilé: Bringing Dynamic Langua... to Static Environ... |
| 10:30 | | | I'm Luvin' It | | | | | | | | | | | | | |
| 10:45 | | | FluxBB | Apache CloudSt... features and tools | | | Why you no host? | | | | Hardening MySQL | | | | | |
| 11:00 | | | | | | | | | OpenSIPS at the core of a distribu... and maintai... VoIP platform | Widget Layout, tutorial | | | | | | |
| 11:15 | | | EDE, a light desktop environ... | Getting Started Hacking on OpenNe... | | | | KDE libraries for Qt applicat... develop... | | | Automa... MySQL failover with MHA: getting started & moving past its quirks | | | Continu... Integrat... and Testing in Games | | |
| 11:30 | Firefox OS | Static site generat... for the masses | Managing your metal flexibly | | Enabling better device interact... with acceler... | Can Linux network configur... suck less? | A method for distribu... applicat... independ... from the distro | | | Debuggi... calc filter problems | | Ada on Android | Event Triggers | | State of Pharo | LPI Exam Session 3 |
| 11:45 | | | Do you want to measure your project? | Enabling cloud for e-Science with OpenNe... | | | | | MSRP and SIP over WebSoc... in | How to debug Writer file format | Trouble... MySQL perform... | | | | | |

github.com/fosdem/website

github.com/fosdem/website



SSGs advantages

SSGs advantages

- ▶ More than Markdown + layout → HTML!

SSGs advantages

- ▶ More than Markdown + layout → HTML!
- ▶ Satisfactory in *many* cases

SSGs advantages

- ▶ More than Markdown + layout → HTML!
- ▶ Satisfactory in *many* cases
- ▶ Dynamic with JavaScript

SSGs advantages

- ▶ More than Markdown + layout → HTML!
- ▶ Satisfactory in *many* cases
- ▶ Dynamic with JavaScript
- ▶ Usable for publishing your data

Lessons Learnt

Internal DSLs

Internal DSLs

```
compile '/about/' do  
  filter :kramdown  
end
```


Internal DSLs – side effects

```
compile '/about/' do
  filter :kramdown

  File.open('output/stuff.txt', 'w') do |f|
    f.write(item.raw_content)
  end
end
```

Internal DSLs – mutability

```
compile '/articles/*/' do
  unless @item[:layout]
    @item[:layout] = 'article'
  end

  layout @item[:layout]
end
```

Speed

Incremental compilation

Incremental compilation

- ▶ Recompile only what's necessary

Incremental compilation

- ▶ Recompile only what's necessary
- ▶ $\Omega(\#items) \rightarrow \Omega(\#changed-items)$

Incremental compilation

- ▶ Recompile only what's necessary
- ▶ $\Omega(\text{\#items}) \rightarrow \Omega(\text{\#changed-items})$
- ▶ Scenarios

Incremental compilation

- ▶ Recompile only what's necessary
- ▶ $\Omega(\text{\#items}) \rightarrow \Omega(\text{\#changed-items})$
- ▶ Scenarios
 - Single changed page

Incremental compilation

- ▶ Recompile only what's necessary
- ▶ $\Omega(\text{\#items}) \rightarrow \Omega(\text{\#changed-items})$
- ▶ Scenarios
 - Single changed page
 - Changed page that is included elsewhere

Incremental compilation

- ▶ Recompile only what's necessary
- ▶ $\Omega(\#items) \rightarrow \Omega(\#changed-items)$
- ▶ Scenarios
 - Single changed page
 - Changed page that is included elsewhere
 - Changed rules

Pick the right libraries

Pick the right libraries

- ▶ RubyPants, Typogruby

Pick the right libraries

- ▶ RubyPants, Typogruby
Winner: Fast Aleck

Pick the right libraries

- ▶ RubyPants, Typogruby
Winner: Fast Aleck
- ▶ pygmentize, pygments.rb

Pick the right libraries

- ▶ **RubyPants, Typogruby**
Winner: Fast Aleck
- ▶ **pygmentize, pygments.rb**
Winner: pygments.rb

ddfreyne opened this pull request 8 months ago

Edit

Use `pygments.rb` instead of `pygmentize`

No one is assigned

No milestone

This pull request lets the site use `pygments.rb` instead of `pygmentize` for syntax coloring. `pygments.rb` support is new in nanoc 3.3, so the Gemfile has been updated to make use of the latest nanoc version.

This change cuts down site compilation speed quite significantly.

Before: 294.59s

After: 2.54s

in nanoc 3.3, so the Gemfile ha

This change cuts down site cor

Before: 294.59s

After: 2.54s

Future Work

Future Work

Future Work

- ▶ Improving speed (Parallelisation)

Future Work

- ▶ Improving speed (Parallelisation)
- ▶ Supporting huge sites

Future Work

- ▶ Improving speed (Parallelisation)
- ▶ Supporting huge sites
- ▶ Writing great documentation

Future Work

- ▶ Improving speed (Parallelisation)
- ▶ Supporting huge sites
- ▶ Writing great documentation
- ▶ Building an active contributors' community

Future Work

- ▶ Improving speed (Parallelisation)
- ▶ Supporting huge sites
- ▶ Writing great documentation
- ▶ Building an active contributors' community
- ▶ GUI?

Recap

SSGs...

SSGs...

- ▶ have big obvious advantages

SSGs...

- ▶ **have big obvious advantages**
fast, secure, easy to deploy, ...

SSGs...

- ▶ **have big obvious advantages**
fast, secure, easy to deploy, ...
- ▶ **have big non-obvious advantages**

SSGs...

- ▶ **have big obvious advantages**
fast, secure, easy to deploy, ...
- ▶ **have big non-obvious advantages**
metadata, data sources, checks, ...

SSGs...

- ▶ **have big obvious advantages**
fast, secure, easy to deploy, ...
- ▶ **have big non-obvious advantages**
metadata, data sources, checks, ...
- ▶ **turn your web site into an open-source project!**

SSGs...

- ▶ **have big obvious advantages**
fast, secure, easy to deploy, ...
- ▶ **have big non-obvious advantages**
metadata, data sources, checks, ...
- ▶ **turn your web site into an open-source project!**
share, learn, collaborate





WWW nanoc.ws

GITHUB github.com/nanoc

IRC [#nanoc](https://freenode.net)

MAIL denis.defreyne@stoneship.org

TWITTER [@ddfreyne](https://twitter.com/ddfreyne)

GITHUB github.com/ddfreyne

Extra Slides!

Preprocessing

```
preprocess do
  ts = Set.new(@items.map { |i| i[:tags] }.flatten)

  ts.each do |tag|
    @items << Nanoc::Item.new(
      "", { :tag => tag }, "/tags/#{tag}/")
  end
end
```