

Static Sites with *Middleman*

Part 1

Static Sites

What are static sites?

HTML, CSS, & JS without a Web Application Server

What are static site generators?

Tools for Programming Static Sites

Why are they useful?

Benefits for Both Development & Deployment

Development

- Templating
- Content Models
- Asset Pipeline
- Extensibility

Deployment

- Simplicity
- Speed
- Caching
- Security

When to use a static site generator?

Depends on the Content

What about dynamic features?

What are the problems?

- Less Mature Ecosystem than a CMS Like WordPress
- Not Designed for Non-Technical Users

Who is using static site generators?

- Google
- Vox Media
- MailChimp
- Sphero
- GitHub
- Carrot - Roots
- Instrument - Middleman

Static Site Generators

The screenshot shows the StaticGen website interface. At the top, the title "StaticGen" is displayed in large white font, followed by the subtitle "Top Open-Source Static Site Generators". Below this is a "SHARE" button. The main content area features a navigation bar with "All languages" and "Sorted by stars" dropdowns, and links for "About StaticGen" and "The Rules". The main content displays four generator cards: Jekyll, GitBook, Hugo, and Hexo. Each card includes the generator's name, website URL, a table of statistics (stars, pull requests, issues), a brief description, and technical details like language, templates, and license.

Generator	Website	Stars	Pull Requests	Issues	Description	Language	Templates	License
Jekyll	jekyllrb.com	27598	6112	125	A simple, blog-aware, static site generator.	Ruby	Liquid	MIT
GitBook	www.gitbook.com/	13502	1647	595	A modern publishing toolchain. Simply taking you from ideas to finished, polished books.	JavaScript	Jinja2	APL 2.0
Hugo	gohugo.io/	13285	2046	360	A Fast and Flexible Static Site Generator.	Go	Go Templates	APL 2.0
Hexo	hexo.io/	13016	2062	309	Hexo is a fast, simple and powerful blog framework.	JavaScript	EJS, Swig	MIT

Part 2

Middleman

Middleman

Search in the documentation

Support

Contribute

Docs



**Build static websites
with an easy-to-use framework**

Middleman is a static site generator using all the shortcuts and tools in modern web development.

Middleman

- Built with Ruby
- Templating Engine - Ruby (ERb)
- Content - Site Map Resource & Collections
- Asset Pipeline - Sprockets or External Pipeline
- Extending - Extensions Available

Development Cycle

```
$ middleman init NEW_MIDDLEMAN_PROJECT_NAME
```

```
OR middleman init NEW_MIDDLEMAN_BLOG_NAME --template=blog
```

```
$ middleman server
```

```
$ middleman console
```

```
$ middleman build
```

Directory Structure

```
mymiddlemansite/  
+-- Gemfile  
+-- Gemfile.lock  
+-- config.rb  
+-- build  
+-- data  
+-- lib  
+-- source  
    +-- images  
        | +-- background.png  
        | +-- middleman.png  
    +-- index.html.erb  
    +-- javascripts  
        | +-- all.js  
    +-- layouts  
        | +-- layout.erb  
    +-- stylesheets  
        +-- all.css  
        +-- normalize.css
```


Templating

1. Layout & Partials
2. Ruby Mixed with HTML

Ruby Code Mixed with HTML

- ERb Enables Ruby Directly Inside HTML
- Ruby Added Between the `<% & %>` Tags
- Code Inside the `<%= %>` Tags Substituted Into the Document

Layouts & Partials

- Layouts `<%= yield %>` to the Current Page Ex. `index.html`
- Partials Insert HTML Snippets Ex. `<%= partial "nav" %>`

Creating Placeholder Text

July, 18 2016

Williamsburg DIY small batch, tousled bitters vice meh nesciunt neutra pabst. Ad tempor cold-pressed, quis tofu ethical cliché. Polaroid squid helvetica sartorial, jean shorts pariatur meggings. Proident readymade assumenda, kombucha yuccie portland...

[Read more...](#)

Why I Decided to Learn Software Development

May, 14 2016

My journey to the Learn Verified program and learning software development has really been a journey of many steps. I've always enjoyed using and learning about technology, so technology has always been a part of my life. However, software development...

[Read more...](#)

Middleman Blog with GitHub Pages

February, 4 2016

Middleman is a static site generation framework with an available blog extension package. I've tried to set up a blog with Middleman in the past, but I've always had trouble with deploying to GitHub pages. To me it's not as straight forward as it should be, but I think with a few tips it can easily be done. Here are the steps that worked for me.

[Read more...](#)

Example:

snsavage.com

Recent Articles

1. [Creating Placeholder Text](#) Jul 18
2. [Why I Decided to Learn Software Development](#) May 14
3. [Middleman Blog with GitHub Pages](#) Feb 4

Tags

1. [middleman \(1\)](#)
2. [github \(1\)](#)
3. [github pages \(1\)](#)
4. [tutorial \(1\)](#)

By Year

1. [2016 \(3\)](#)

source/layout.erb

```
<!doctype html>
<html>
  <head>
    <%= partial "head" %>
  </head>
  <body>
    <div id="container">
      <header>
        <%= partial "nav" %>
      </header>
      <main>
        <div id="main" role="main">
          <%= yield %>
        </div>
        ...
      </main>
      <footer>
        <%= partial "footer" %>
      </footer>
    </div>
  </body>
</html>
```

source/partials/_nav.erb

```
<nav class="clearfix hidden-nav-menu">
  <h1>snsavage.com</h1>
  <h1 class="hidden-nav-trigger">
    <a href="#">
      <i class="fa fa-bars fa-lg" aria-hidden="true"></i>
    </a>
  </h1>
  <ul class="nav-menu">
    <%= partial "nav_links" %>
  </ul>
</nav>
```

source/partials/_nav_links.erb

```
<li><%= link_to "Home", "/" %></li>
```

```
<li><%= link_to "About", "about.html" %></li>
```

```
<li><%= link_to "Contact", "contact.html" %></li>
```

build/index.html

```
<!doctype html>
<html>
<head>
  ...
</head>
<body>
  <div id="container">
    <header>
      <nav class="clearfix hidden-nav-menu">
        <h1>snsavage.com</h1>
        <h1 class="hidden-nav-trigger">
          <a href="#"><i class="fa fa-bars fa-lg" aria-hidden="true"></i></a>
        </h1>
        <ul class="nav-menu">
          <li><a href="/">Home</a></li>
          <li><a href="/about.html">About</a></li>
          <li><a href="/contact.html">Contact</a></li>
        </ul>
      </nav>
    </header>
    ...
  </div>

```


Creating Placeholder Text

July, 18 2016

Williamsburg DIY small batch, tousled bitters vice meh nesciunt neutra pabst. Ad tempor cold-pressed, quis tofu ethical cliché. Polaroid squid helvetica sartorial, jean shorts pariatur meggings. Proident readymade assumenda, kombucha yuccie portland...

[Read more...](#)

Why I Decided to Learn Software Development

May, 14 2016

My journey to the Learn Verified program and learning software development has really been a journey of many steps. I've always enjoyed using and learning about technology, so technology has always been a part of my life. However, software development...

[Read more...](#)

Middleman Blog with GitHub Pages

February, 4 2016

Middleman is a static site generation framework with an available blog extension package. I've tried to set up a blog with Middleman in the past, but I've always had trouble with deploying to GitHub pages. To me it's not as straight forward as it should be, but I think with a few tips it can easily be done. Here are the steps that worked for me.

[Read more...](#)

Example:

snsavage.com

Recent Articles

1. [Creating Placeholder Text](#) Jul 18
2. [Why I Decided to Learn Software Development](#) May 14
3. [Middleman Blog with GitHub Pages](#) Feb 4

Tags

1. [middleman \(1\)](#)
2. [github \(1\)](#)
3. [github pages \(1\)](#)
4. [tutorial \(1\)](#)

By Year

1. [2016 \(3\)](#)

Generating Lists of Articles

```
<ol>
  <% blog.articles[0...10].each do |article| %>
    <li>
      <%= link_to article.title, article %>
      <span><%= article.date.strftime('%b %e') %></span>
    </li>
  <% end %>
</ol>
```

Titles Based on Page Content

```
<title>  
  snsavage.com  
  <%= ' - ' + current_article.title unless current_article.nil? %>  
</title>
```

Middleman Content Model

Pages, Frontmatter, & Local Data

Pages

- Add a New HTML File to Add Pages
- Adding the *.erb Extension to Include Ruby
- Add a Blog Article with \$ middleman articles
ARTICLE_TITLE

Frontmatter Format

title: Middleman Blog with GitHub Pages

author: Scott Savage

date: 2016-02-04 02:40 UTC

tags: middleman, github, github pages, tutorial

published: true

Using Frontmatter Data

```
<% wrap_layout :layout do %>
  <article>
    <h2><%= current_page.data.title %></h2>
    <h3><%= current_page.date.strftime( '%B, %e %Y' ) %></h3>
    <h4>By <%= current_page.data.author %></h4>
    <%= yield %>
  </article>
<% end %>
```

Local Data

- Stored in the `data` Directory
- Files in the `.yaml`, `.yml`, or `.json` Formats
- Accessed by the `data` Object

Helper Methods

Simplify Common HTML Tasks

- Links - `link_to`
- Tags - `tag`, `content_tag`
- Forms - `form_tag`, `label_tag`, `submit_tag`
- Formatting - `pluralize`, `word_wrap`, `truncate`

Middleman Asset Pipeline

- Rails Asset Pipeline (Sprockets) - v3
- External Pipeline - v4

Middleman Extensions

- `middleman-blog` - Middleman Blogging Extension
- `middleman-syntax` - Code Syntax Highlighting
- `contentful_middleman` - Contentful CMS API
- `middleman-deploy` - Deploy Using sync, ftp, sftp, or git
- `middleman-disqus` - Integrate Disqus Into Middleman

Part 3

Example Workflow

snsavage.com

Home

About

Portfolio

Contact

The Making of RetroCasts

October 14, 2016

RetroCasts provides a command line utility (cli) for quickly accessing episode data from RailsCasts.com. It provides a quick and convenient way to search the site without leaving the terminal. This post discusses the process for making the RetroCasts gem. The first major project for the Flatiron School's Online Web Developer program is the "CLI Data Gem Project." So far the program has had several other larger projects, but this is the first project where a test suite is not provided as a framework.

[Read more...](#)

Static Sites with Middleman

August 26, 2016

In August 2016 I gave a talk at the Lancaster Front-End Meetup about static web sites and the static site generator Middleman.

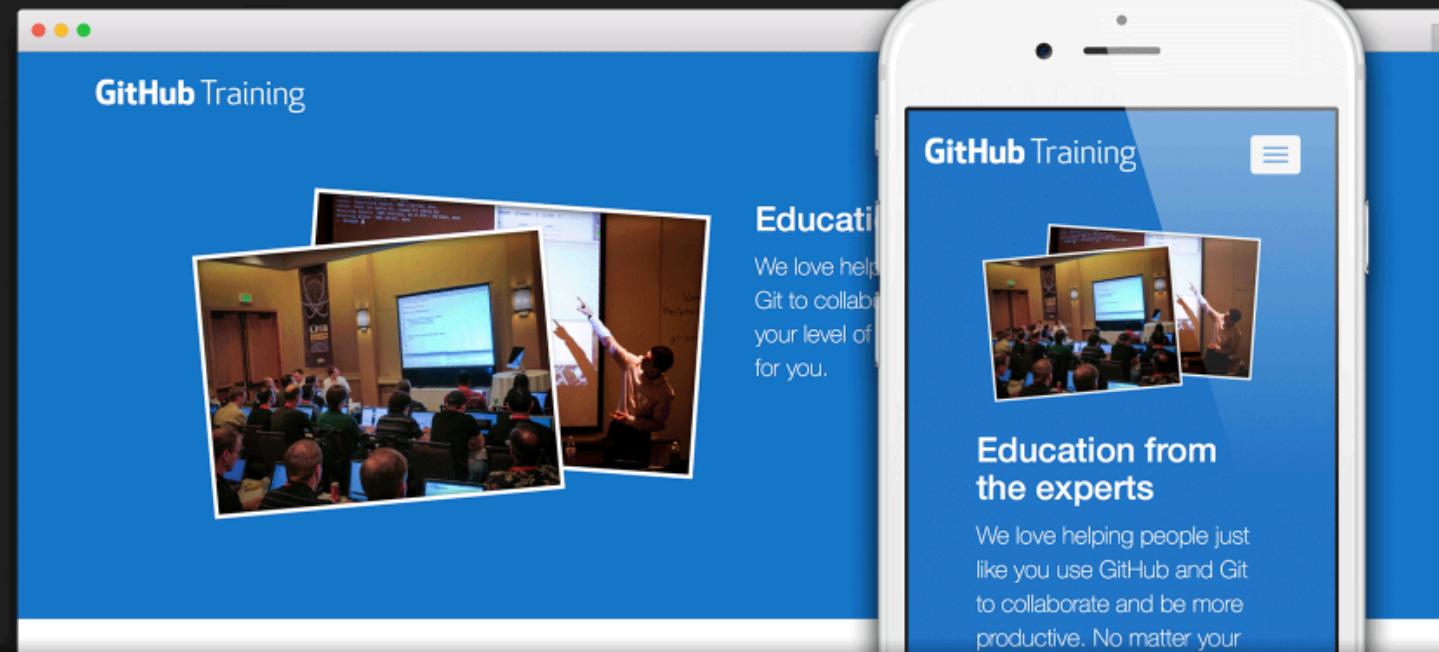
[Read more...](#)

GitHub Pages

GitHub Pages

Websites for you and your projects.

Hosted directly from your [GitHub repository](#). Just edit, push, and your changes are live.



Netlify



netlify

[Features](#) [Pricing](#) [Docs](#) [About](#) [Blog](#) [Support](#) [🔍](#)

[Log In](#)

[Sign Up](#)



Write frontend code. Push it.
We handle the rest.

CDN, Continuous deployment, 1 click HTTPS, and all the services you need. From instant prototypes to huge global enterprises.

[Get Started with Netlify](#)



What if a website has non-technical users?

Contentful

[Features](#)[Customers](#)[Pricing](#)[Docs & Help](#)[About us](#)[Contact](#)[Log in](#)[Try for free](#)

Like a CMS... without the bad bits.

Contentful is a content management developer platform with an API at its core.

```
$ curl 'https://cdn.contentful.com/spaces/developer_bookshelf/entries?
access_token=0b7f6x59a0&content_type=book'

{
  "total" : 2,
  "items" : [
    {
      "fields" : {
        "name" : "An introduction to regular expressions. Volume VI",
        "author" : "Larry Wall",
        "description" : "Now you have two problems."
      }
    },
    {
      "fields" : {
        "name" : "How to manage content in a developer-friendly manner",
        "author" : "Contentful",
        "description" : "Make an API request, get JSON in return."
      }
    }
  ]
}
```

AMONG OUR
CUSTOMERS:

INSTRUMENT



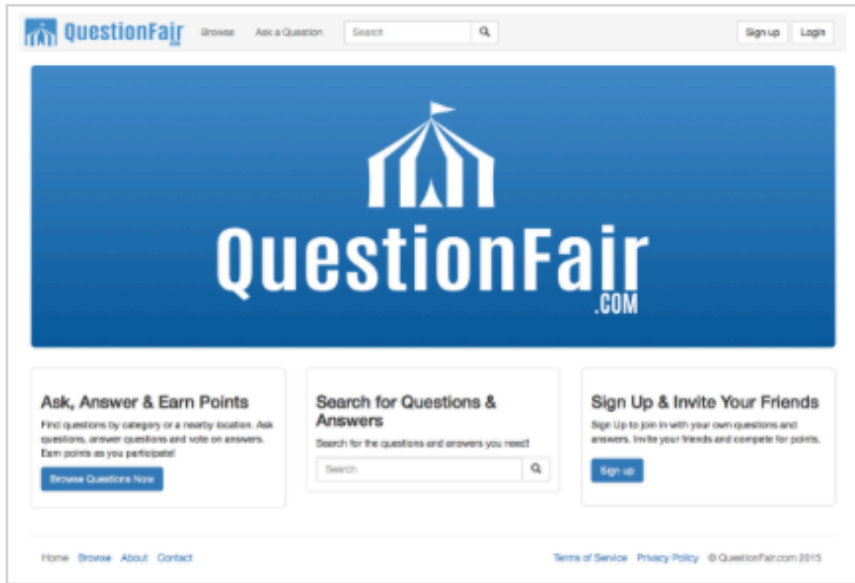
SPECIALIZED



snsavage.com/portfolio.html

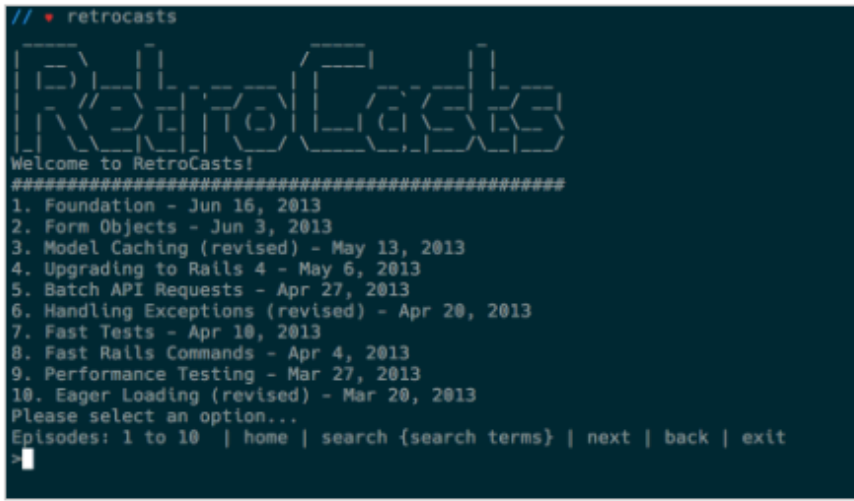
snsavage.com Home About Portfolio Contact

QuestionFair.com



QuestionFair.com is a website for asking questions and getting answers. The website includes a few more advanced features including location tagging, points rewards and friendships with other users. The web app was created as a final academic project for completion of a Master's degree in Information Systems. QuestionFair was built using the Ruby on

RetroCasts



RetroCasts provides a command-line interface for access to metadata from RailsCasts.com. Features include episode lists, episode details, and search.

Find Out More: [GitHub Article](#) [RubyGems](#)

Custom Contentful CMS

The screenshot displays a web-based CMS interface for editing a content item. The top navigation bar includes the site name 'snsavage.com', navigation links for 'Content' and 'Media', and a user profile for 'Scott'. The main editing area is titled 'Untitled' and contains several form fields: 'Name (required)' with a character count of 0 and a 256-character limit; 'Release Date (required)' with a calendar icon; and a 'Description' field with a rich text editor toolbar (including bold, italic, quote, list, link, and media icons) and a 'Preview' tab. Below the description field, there are 'Media' and 'Links' sections, each with buttons to 'Create new asset/link' and 'Add some existing assets/entries'. A right-hand sidebar provides additional controls: 'STATUS' (Draft, with a 'Publish' button and 'Last saved 18 minutes ago' timestamp), 'TRANSLATION' (en-US, with a 'Change' link), 'VERSIONS' (with a 'Compare versions' button and explanatory text), and 'USERS' (showing 'No other users online').

source/portfolio.html.erb

```
<div class="portfolio">
  <% data.portfolio.project.each do |id, project| %>
    <div class="project">

      <h2><%= project["name"] %></h2>
      ...
      <div class="project-description">
        <p><%= project["description"] %></p>

        <div class="project-links">
          <h4>Find Out More:</h4>
          <% if project.links %>
            <% project.links.each do |link| %>
              <%= link_to link.name, link.url %>
            <% end %>
          <% end %>
        </div>
      </div>
    </div>
  </div>
<% end %>
</div>
```

Publishing New Content

- Clicking Publish in Contentful Calls Netlify Webhook
- Content is Updated with `$ middleman contentful`

Netlify Admin

The screenshot shows the Netlify Admin dashboard for the SNSAVAGE site. At the top, the Netlify logo and navigation links (Sites, About, Documentation) are visible, along with the user email 'snsavage@gmail.com'. The main header displays 'SNSAVAGE' and a link to 'ALL SITES'. A navigation bar contains 'Settings', 'Deploys', 'Forms', 'Access', 'Notifications', and 'HTTPS'. Below this, an 'Upgrade' button is present. The main content area is split into two columns. The left column shows a preview of the SNSAVAGE website with three blog posts: 'The Making of RetroCasts' (dated October 14, 2016), 'Static Sites with Middleman' (dated August 26, 2016), and 'My Path to Software Development' (dated August 25, 2016). The right column displays site configuration options: 'Name: snsavage edit', 'Domain: www.snsavage.com edit', and 'Password Protection: Available on paid plans Upgrade'. At the bottom, the 'Link to Git' section shows 'Settings for continuous deployment from git.' and lists: 'Repository: https://github.com/snsavage/snsavage-netlify Edit', 'Branch: master', 'Build Cmd: middleman contentful && middleman build', and 'Public folder: build/'. A chat icon is located in the bottom right corner.

netlify Sites About Documentation snsavage@gmail.com

SNSAVAGE

← ALL SITES

Settings Deploys Forms Access Notifications HTTPS

Upgrade

snsavage.com Home About Portfolio Contact

The Making of RetroCasts

October 14, 2016

RetroCasts provides a command line utility (cli) for quickly accessing episode data from RailsCasts.com. It provides a quick and convenient way to search the site without leaving the terminal. This post discusses the process for making the RetroCasts gem. The first major project for the Flatiron School's Online Web Developer program is the "CLI Data Gem Project." So far the program has had several other larger projects, but this is the first project where a test suite is not provided as a framework. [Read more.](#)

Static Sites with Middleman

August 26, 2016

In August 2016 I gave a talk at the Lancaster Front-End Meetup about static web sites and the static site generator Middleman. [Read more.](#)

My Path to Software Development

August 25, 2016

As part of the Flatiron School's Online Web Developer Program, the students have been asked to write regular blog posts. The first topic asks why we decided to learn software development. I think that my reason is best told by a story about why I joined the program. [Read more.](#)

Name: snsavage [edit](#)

Domain: www.snsavage.com [edit](#)

Password Protection:
Available on paid plans [Upgrade](#)

Link to Git

Settings for continuous deployment from git.

Repository: <https://github.com/snsavage/snsavage-netlify> [Edit](#)

Branch: master

Build Cmd: middleman contentful && middleman build

Public folder: build/

Resources

- [Middleman](#)
- [Smashing Magazine - Why Static Website Generators Are The Next Big Thing](#)
- [Smashing Magazine - Static Website Generators Reviewed](#)
- [Static Webtech Meetup - Dynamic to Static](#)
- [Netlify](#)
- [Contentful](#)
- [YouTube - Middleman External Pipeline + Gulp](#)

About Me:

- E-Mail: snsavage@gmail.com
- Website: snsavage.com
- GitHub: github.com/snsavage



- [Flatiron School](#)
- learn.co/with/snsavage

Questions?