Jekyll

From ArchWiki

Jekyll (http://jekyllrb.com/) is a simple, blog aware, static site generator written in Ruby and developed by GitHub co-founder Tom Preston-Werner (http://tom.preston-werner.com/). It takes a template directory (representing the raw form of a website), runs it through Textile or Markdown and Liquid converters, and spits out a complete, static website suitable for serving with Apache or your favorite web server. It is the engine behind GitHub Pages (http://pages.github.com/).

Werner announced the release of Jekyll on his website (http://tom.preston-werner.com/2008/11/17/blogging-like-a-hacker.html) on November 17, 2008.

Contents

Installation

Jekyll can be installed in Arch Linux with the RubyGems package manager or using the applicable packages in the AUR. Both methods require the Ruby package in the official repositories to be installed.

RubyGems (recommended)

Note: RubyGems 1.8 and above are displaying numerous uncritical warnings (https://github.com/rspec/rspec-core/issues/345).

The best way to install Jekyll is with RubyGems, a package manager for the Ruby programming language. RubyGems is installed alongside the `ruby (https://www.archlinux.org/packages/?name=ruby)` package, which is located in the official repositories. Jekyll can then be installed for all users on the machine using the `gem` command as root. Alternative installation methods are available on the Ruby page.

Before installing Jekyll make sure to update RubyGems.

```
$ gem update
```

Then install Jekyll using the `gem` command.

```
$ gem install jekyll
```

See Ruby#RubyGems for more information on Gem management in Arch.

AUR (alternate)

Alternately, `ruby-jekyll (https://aur.archlinux.org/packages/ruby-jekyll/)` can be installed from the AUR.

Rubygems binary repository

Install *ruby-jekyll* from the unofficial quarry repository.

Select a markup language

There are numerous different markup languages that are used to define text-to-HTML conversion tools. Jekyll has two defaults; Textile and Markdown (http://daringfireball.net/projects/markdown/). Implementations of both are required as dependencies of Jekyll.

Textile

Textile is a markup language used by Jekyll.

Note: RedCloth, a module for using the Textile markup language in Ruby, fails to install with gcc 4.6.0 (see: RedCloth Ticket 215 (http://jgarber.lighthouseapp.com/projects/13054/tickets/215-native-ext-compilation-failure) and 219 (http://jgarber.lighthouseapp.com/projects/13054/tickets/219-427-installation-issue-on-arch-linux-x64)). It is recommended that you install the current stable version 4.2.2 by `gem install RedCloth --version 4.2.2`.

Markdown

Markdown (http://daringfireball.net/projects/markdown/) is a markup language and text-to-HTML conversion tool developed in Perl by John Gruber (http://daringfireball.net/). A Perl and a Python implementation of Markdown can be found in the official repositories, while numerous other implementations are available in the AUR (https://aur.archlinux.org/packages.php?O=0&K=markdown&do_Search=Go). The default implementation of Markdown in Jekyll is Maruku (http://maruku.rubyforge.org/index.html).

Additionally, it has been implemented in C as Discount (http://www.pell.portland.or.us/~orc/Code/discount/) by David Parsons (http://www.pell.portland.or.us/~orc/) and a Ruby extension was written by Ryan Tomayko (http://tomayko.com/) as RDiscount (https://github.com/rtomayko/rdiscount). You can install RDiscount with Rubygems as root **or** through `ruby-rdiscount (https://aur.archlinux.org/packages/ruby-rdiscount/)` AUR package.

```
# gem install rdiscount -s http://gemcutter.org
```

Then add the following line to your `_config.yml`.

```
markdown: rdiscount
```

If you are unfamiliar with Markdown, Gruber's website (http://daringfireball.net/projects/markdown/basics) presents an excellent introduction. Additionally, you can try out Markdown using Gruber's online conversion tool (http://daringfireball.net/projects/markdown/dingus).

Configuration

A default Jekyll directory tree looks like the following, where "." denotes the root directory of your Jeykll generated website.

```
.
|-- _config.yml
|-- _layouts
|   |-- default.html
|   `-- post.html
|-- _posts
|   |-- 2010-02-13-early-userspace-in-arch-linux.textile
|   `-- 2011-05-29-arch-linux-usb-install-and-rescue-media.textile
|-- _site
`-- index.html
```

A default file structure is available from Daniel McGraw's Jekyll-Base (https://github.com/danielmcgraw/Jekyll-Base) page on GitHub.

Note: McGraw has also setup a more extensive default file structure on GitHub (https://github.com/danielmcgraw/danielmcgraw.com.git).

The `_config.yml` file stores configuration data. It includes numerous configuration settings, which may also be called as flags. Full explanation and a default configuration can be found on GitHub (https://github.com/mojombo/jekyll/wiki/Configuration).

Once you have configured your `_config.yml` to your liking you need to create the files that will be processed by Jekyll to generate the website.

Usage

Next you need to create templates that Jekyll can process. These templates make use of the Liquid templating system to input data. For a full explanation check GitHub (https://github.com/mojombo/jekyll/wiki/template-data).

Additionally, each file besides `/_layouts/layout.html` requires a YAML Front Matter (https://github.com/mojombo/jekyll/wiki/yaml-front-matter) heading.

Create index layout

This is a basic template for your `index.html`, which is used to render your website's index page.

```
---
layout: layout
title: Jekyll Base
---

<div class="content">
  <div class="related">
    <ul>
      {% for post in site.posts %}
      <li>
 <span>{{ post.date | date: "%B %e, %Y" }}</span> <a href="{{ post.url }}">{{ post.title }}</a>
      </li>
      {% endfor %}
    </ul>
  </div>
</div>
```

Create general website layout

This is a basic template for your website's general layout. It will be referenced in the YAML Front Matter blocks of each file (see: Creating a Post).

_layouts/layout.html

```
<!DOCTYPE HTML>

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <meta name="author" content="Your Name" />
    <title>{{ page.title }}</title>
  </head>
  <body>
    <header>
      <h1><a href="/">Jekyll Base</a></h1>
    </header>
    <section>
      {{ content }}
    </section>
  </body>
</html>
```

Create post layout

This is a basic template for each of your posts. Again, this will be referenced in the YAML Front Matter blocks of each file (see: Creating a Post).

_layouts/post.html

```
---
layout: layout
title: sample title
---

<div class="content">
  <div id="post">
    <h1>{{ post.title }}</h1>
    {{ content }}
  </div>
</div>
```

Creating a post

The content of each blog post will be contained within a file inside of the `_posts` directorys. To use the default naming convention each file should be saved with the year, month, date, post title and end with the *.md or *.textile depending on the markup language used (e.g. `2010-02-13-early-userspace-in-arch-linux.textile`). The date defined in the filename will be used as the published date in the post. Additionally, the filename will be used to generate the permalink (i.e. /categories/year/month/day/title.html). To use an alternate permalink style or create your own review the explanation on GitHub (https://github.com/mojombo/jekyll/wiki/Permalinks).

Test

To generate a static HTML website based on your Textile or Markdown documents run `jekyll`. To simultaneously test the generated HTML website run Jekyll with the `--serve` flag.

```
$ jekyll serve
```

or if you want jekyll to watch for file changes

```
$ jekyll serve --watch
```

It is recommended to define server options in your `_config.yml`. The default will start a server on port 4000, which can be accessed in your web browser at `localhost:4000`.

See also

- YAML
- Textile
- Installation Tutorial (http://danielmcgraw.com/2011/04/14/The-Ultimate-Guide-To-Getting-Started-With-Jekyll-Part-1/) by Daniel McGraw
- Configuration Tutorial (http://danielmcgraw.com/2011/04/18/The-Ultimate-Guide-To-Getting-Started-With-Jekyll-Part-2/) by Daniel McGraw
- Jekyll vs. Hyde (http://philipm.at/2011/0507/) by Philip Mateescu
- Websites created with Jekyll can be found on GitHub (https://github.com/mojombo/jekyll/wiki/sites)
- **Required software:**
  - http://redcloth.org - RedCloth
  - http://www.liquidmarkup.org/ - Liquid
  - http://classifier.rubyforge.org/ - Classifier
  - http://maruku.rubyforge.org/maruku.html - Maruku
  - http://pygments.org/ - Pygments
  - http://rubygems.org/gems/directory_watcher - Directory Watcher

Retrieved from "https://wiki.archlinux.org/index.php?title=Jekyll&oldid=339996"
Category: Web Server

- This page was last modified on 14 October 2014, at 19:44.
-