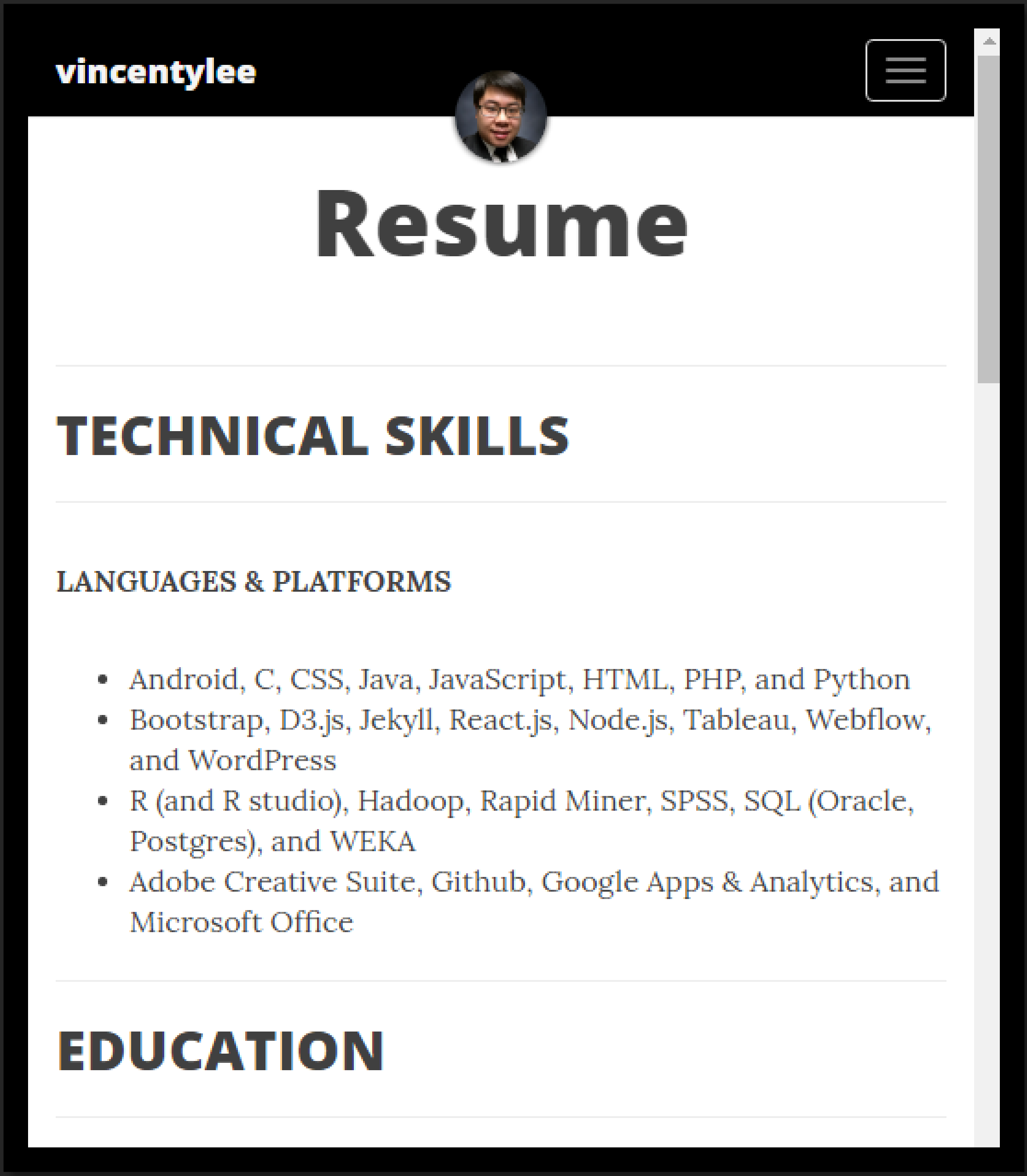
How to Build A Free Website Portfolio Using Jekyll



Resume

TECHNICAL SKILLS

LANGUAGES & PLATFORMS

- Android, C, CSS, Java, JavaScript, HTML, PHP, and Python
- Bootstrap, D3.js, Jekyll, React.js, Node.js, Tableau, Webflow, and WordPress
- R (and R studio), Hadoop, Rapid Miner, SPSS, SQL (Oracle, Postgres), and WEKA
- Adobe Creative Suite, Github, Google Apps & Analytics, and Microsoft Office

EDUCATION

By Vincent Lee

Table of Contents

Abstract

Why build a website portfolio using Jekyll?

I was always fascinated by web development, but it took me a long time to discover where to start and how to deploy a website. By the time I figured it out, I realized that owning and maintaining your own personal website was COSTLY. Even if you use services such as *Squarespace* you're still paying an upwards of hundreds of dollars for what they would call a "basic template" that everyone has.

Today I am going to teach you how to build a FREE static website portfolio using Jekyll and Git pages. The reason why I encourage you to make website is because you can start a blog, showcase your work, and even use it as a student portfolio.

A resume has only a limited amount of space and recruiters take 30 seconds to glance over your resume. You cannot expect them to be well-informed and know all your shining achievements. Instead, you want to put everything down into a portfolio that is easy to look at and completely eye catching.

My resume and *LinkedIn* are great resources for people to get in touch with me, but they do not help me individualize myself from the rest of the world. My Resume and *LinkedIn* use the same generic format that the platform is intended to, but no one looks at it because it gets lost in the ocean of other similar pages.

A portfolio website become my unique identifier so long as people find the link, URL, or even the name of my site. They can go visit it (e.g. vincentylee.com)! If people are interested in project or something you've done on your resume, they will take the time to "type in" your website in their office or on their smartphone.

What is Jekyll

By Wikipedia's definition:

"Jekyll is a simple, blog-aware, static site generator for personal, project, or organization sites. Written in Ruby by Tom Preston-Werner, GitHub's co-founder, it is distributed under an open source license."

My personal definition:

- Jekyll is a static site generator that can be used as a blog, a portfolio, or an informational website.
- Jekyll is free and open-source. It's relatively new and simple in nature.
- Jekyll relies on Ruby, Markdown, and Git Pages.

What is a Static Webpage

The definition is taken from *Wikipedia*: "A **static web page** (sometimes called a **flat page/stationary page**) is a web page that is delivered to the user exactly as stored, in contrast to dynamic web pages which are generated by a web application. Consequently, a static web page displays the same information for all users, from all contexts, subject to modern capabilities of a web server to negotiate content-type or language of the document where such versions are available and the server is configured to do so."

- Essentially you have a template + content = collection of HTML pages.
- Basic and easy to develop, but lack dynamic features.
- Does not have a database:
  - No login or user accounts
  - Relies on embedded content or plugins
  - Does not store or collect data

Pros

- Fast and responsive site
- More secure because you have less to worry about
- Less Maintenance and Management
- Free / Low Cost by hosting your static website on Git Pages
- Blog Aware & Markdown

Cons

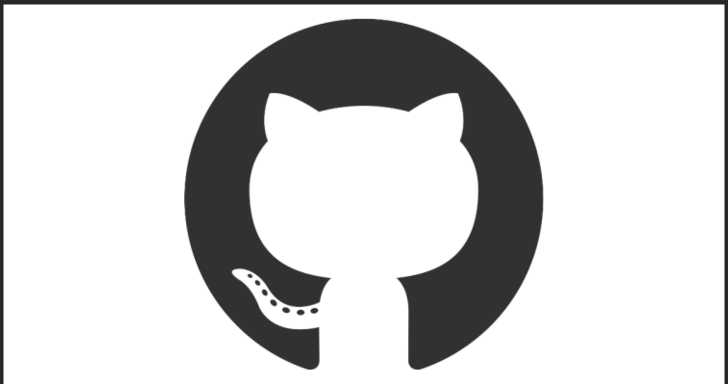
- Does not include a database of dynamic features. In other words, it’s less functional as a website
- Lacks the functionality to incorporate an ecommerce, forums, or file transfer system.
- Does not have enough customization option and lacks proper search engine optimization.

Getting Started

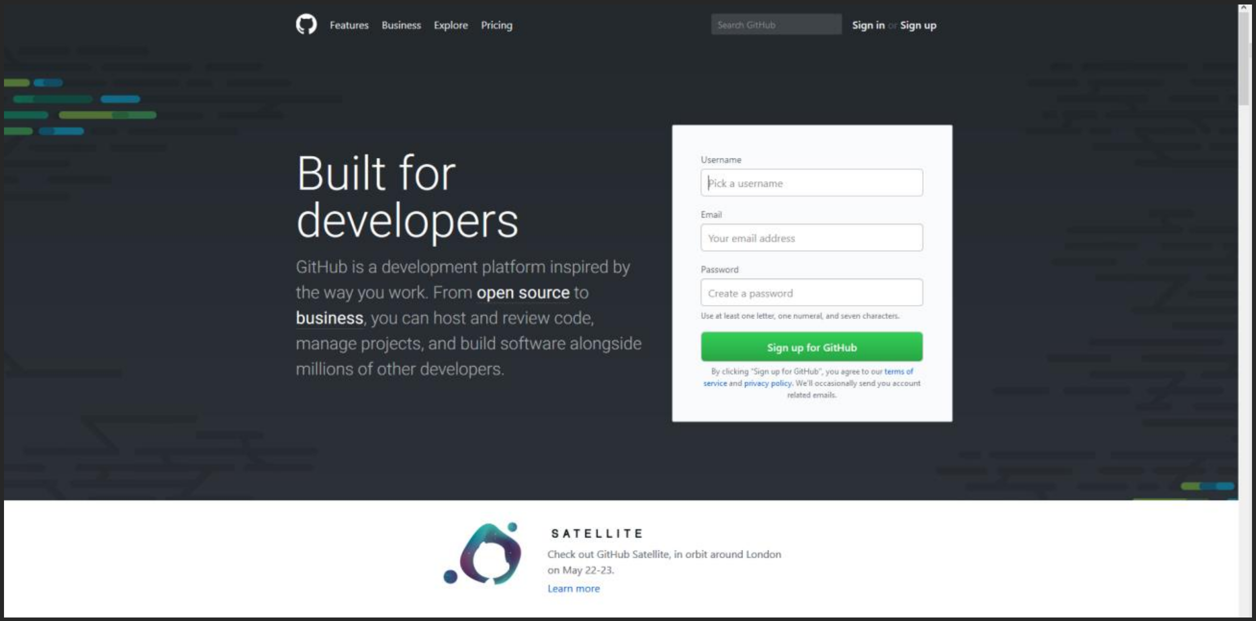
Note: I am unable to help with resolving installation bugs or hardware issues

In this section I will be telling you how to setup and install Jekyll. I would also like you to create a GitHub account. For more information go to the main Jekyll website here.

Creating a Github Account



1. Click the image or this link to create a new GitHub Account: https://github.com/



Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside millions of other developers.

Username
Pick a username

Email
Your email address

Password
Create a password

Use at least one letter, one numeral, and seven characters.

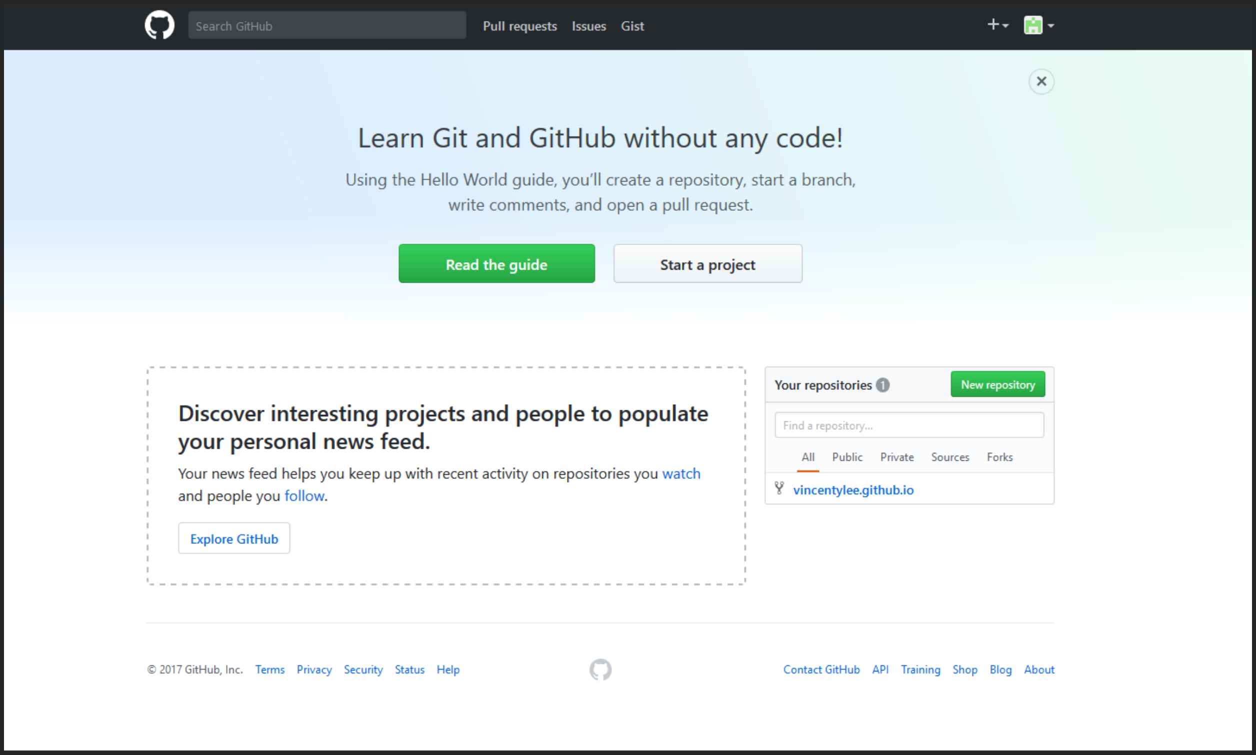
By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We'll occasionally send you account related emails.

SATELLITE

Check out GitHub Satellite, in orbit around London on May 22-23.

Learn more

2. Register or Signup for a new account. Keep in mind, you only get one Git Page per Account.



Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you watch and people you follow.

Your repositories 1

vincentylee.github.io

© 2017 GitHub, Inc.

3. Once you finish signing up, login to make sure everything is okay. For now, leave the page.

Setting up with MACS

1. Installation on Mac is relatively simple compared to the process of installing it on Windows. Find the terminal or your command prompt on your computer and open it. Download Ruby, Jekyll, and Bundler by by writing down this command in the terminal (to make sure it’s installed type e.g. filename -v):

2. Sudo gem install ruby

3. Sudo gem install Jekyll

4. Sudo gem install bundler

Setting up with Windows

1. Find the terminal or your command prompt on your computer and open it. Download Ruby, Jekyll, and Bundler. Follow the exact instruction on these sites:

http://jekyll-windows.juthilo.com (follow step 1 and 2)

http://molevol.altervista.org/blog/install-jekyll-windows/?doing_wp_cron=1493481236.9909179210662841796875

2. Install ruby

3. Gem install Jekyll

4. Gem install bundler

Creating a local server

1. Now let’s create a local server to give you an example of that a plain Jekyll website will look like without any content or template:
2. cd Desktop
3. Jekyll new portfolio
4. cd portfolio
5. Jekyll serve
6. Go to your web browser (e.g. Chrome) and type in “localhost:4000” in the URL

Now that you’ve created a plain Jekyll to make sure everything is working properly, COMPLETELY IGNORE what you’ve just done and close it out.

Picking A Template

Jekyll has an amazing selection of free or open source templates within the public domain that you can choose and download for your webpage. Note: Fancy website templates will have complicated instructions for how to set it up and additional ruby installations.

For Jekyll Templates:

- http://jekyllthemes.org
- https://learn.cloudcannon.com/jekyll-templates/
- Google “Jekyll Themes” or “Jekyll Templates”

Choosing a template

- http://www.jekyllnow.com/
- http://bencentra.com/centrarium/
- http://jekyllthemes.org/themes/freelancer/

These are good examples of templates you can instantly setup by following their EXACT instructions (go to their github page if their demo doesn't have it).

Following Instructions

- Most cases you can just fork the page over in GitHub
- Other times there are specific instructions that require you to download or install multiple packages. Then push it to GitHub via a terminal.

Setting it up in Github

- If you haven't already created a github account

- Once you've created your github account either clone or fork the template into your github account (note: if you haven't synced your github account, it would be easier to fork the project over)

- Now open the project that you've just copied over to you github

- Go over to Settings and change the repository name

- Change it to your username.github.io

- Hooray! You've created your Git Page (you only get 1 Git Page per Account)

Understanding the Layout

Looking at the files

Different Files or Folders

- config.yml: the most important information and setting for your website
- post: where you insert blog post using markdown
- Img: all your images in a one collective folder
- Ignore everything else for now

Creating a Markdown file

- Read your test markdown file that is autogenerated on your website
- Learn to add images, code snippets, links, etc.
- 2016-02-20-file_name.md (have this exact format)

- Date | File Name | Markdown File

Creating and Making Changes

You can start adding your content via:

- Github
- Text Editor
- Or R Studio with Github

It would help if you have some working knowledge of:

- Github
- HTML / CSS
- Markdown
- General Web Dev Skills

Publishing your revisions

- If you have experience with github commands. Just commit and push all your changes online.
- If you don't have experience with github commands (you really should). Go to your github account and make changes there directly
- Check your webpage at www.username.github.io
- Refresh often to see the changes update

Creating A Domain Name

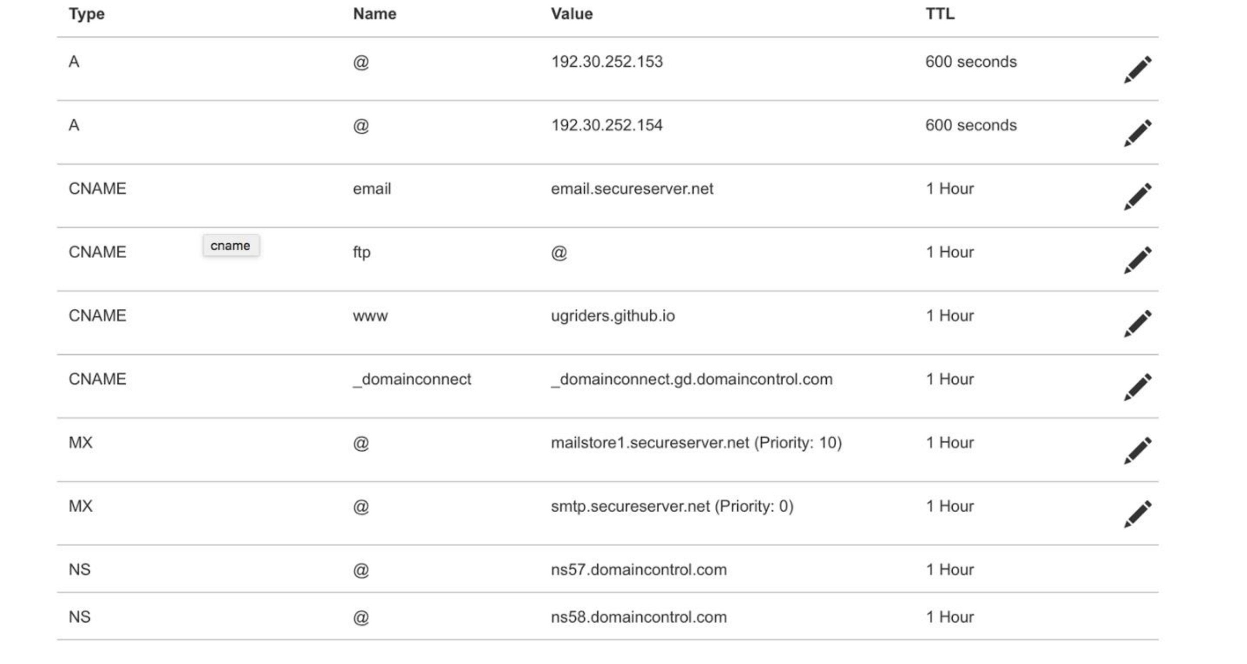
Searching for a domain name

- Everything from this point is completely free.
- Normally you want to change your url link because it's ugly.
- To do that you need to buy a domain name (costs may vary):
    - https://www.godaddy.com/?ci=
- Websites ending with .com, .net, .org are more expensive because they're more recognizable and get better SEO results in google.
- Don't put your actual phone number because of telemarketers

How to buy a domain name

It's relatively simple, once you've picked out a domain name for the right price, create an account and enter in your credit card. You will then be given account information for how you can implement the domain name.

Setting it up to your new website



Type	Name	Value	TTL
A	@	192.30.252.153	600 seconds
A	@	192.30.252.154	600 seconds
CNAME	email	email.secureserver.net	1 Hour
CNAME	ftp	@	1 Hour
CNAME	www	ugriders.github.io	1 Hour
CNAME	_domainconnect	_domainconnect.gd.domaincontrol.com	1 Hour
MX	@	mailstore1.secureserver.net (Priority: 10)	1 Hour
MX	@	smtp.secureserver.net (Priority: 0)	1 Hour
NS	@	ns57.domaincontrol.com	1 Hour
NS	@	ns58.domaincontrol.com	1 Hour

- Go to Domain> Domain Settings> Manage DNS
- Make sure to change the A and CNAME
- If you have troubles go here: http://andrewsturges.com/blog/jekyll/tutorial/2014/11/06/github-and-godaddy.html
- Also go over to your project in your github account
- Create a new file called "CNAME"
- Add your website name in the textbox and commit your changes

You're Done! Sorry for being sloppy at the end. I will revise this pdf the next time I remember to or when I am not busy.