

# NQBH's GitHub Page

Nguyễn Quân Bá Hồng\*

Ngày 22 tháng 5 năm 2022

## Abstract

The construction of NQBH's website.

## Contents

<b>1</b>	<b>Build NQBH's GitHub Page</b>	<b>1</b>
1.1	Getting Started	1
1.2	Previewing Website	2
1.3	Getting Online	16
1.3.1	Uploading changes to GitHub	16
<b>2</b>	<b>NQBH.github.io</b>	<b>16</b>
2.1	Hit Counter	17
2.1.1	JavaScript in Plain English/Mehdi Aoussiad. How to Count Page Views with The Count API	17
2.2	CountAPI	17

## 1 Build NQBH's GitHub Page

**Question 1.1.** *What should I name my personal page/website?*

*Answer.* For authenticity<sup>1</sup> reason, I should use my real name instead of some nickname, which sounds cool now but will turn into bullshit later in the long run, also non-unique. My ear will reject any kind of nickname like that – too childish. Use my real name then. 1st of all, the option NGUYENQUANBAHONG, although unique (Who the hell has the same name as me?), seems too long which even makes me feel lazy to type every time I want to enter my website. NQBH seems to be a good compromise between brevity & uniqueness. I do not think there are a lot of people whose their names share this abbreviation with me. Thus, I choose NQBH for the main part of my website name.

*Then what about the extension?* The commons are .COM, .EDU, .VN, etc. *Which one should I choose?* Using method of elimination, I should not choose .EDU since I will post some “uneducated” contents, however, with some education purposes, which will not yield any contradiction at all. Posting some uneducated contents under a website with .EDU extension seems to contaminate<sup>2</sup> the prestigious label “edu” in various senses. How about .COM, .VN extensions? These seem too common & cost some money monthly/yearly, of course. Hence, I should use some free website builders/hosting with non-annoying extension. GITHUB's personal page seems a good choice for me. □

Next, I follow the following instruction: [Rob Williams's Building an Academic Website](#).

### 1.1 Getting Started

“Most universities these days provide a free option, usually powered by [WordPress](#) (both [WashU](#) & [UNC](#) use WordPress for their respective offerings). While these sites are quick to set up & come with the prestige of a .edu URL, they have several drawbacks that have been extensively written on, e.g.,

- University of California, Berkeley/Townsend Center for the Humanities/[Rochelle Terman. Personal Academic Webpages: How-To's & Tips for a Better Site](#).

---

\*Independent Researcher, Ben Tre City, Vietnam  
e-mail: [nguyenquanbahong@gmail.com](mailto:nguyenquanbahong@gmail.com)

<sup>1</sup>**authenticity** [n] [uncountable] the quality of being genuine or true, or based on fact.

<sup>2</sup>**contaminate** [v] **1.** [usually passive] to make a substance or place dirty or no longer pure by adding a substance that is dangerous or carries disease; **2. contaminate something** to influence people's ideas or attitudes in a bad way.

- *Martin Lea. [Personal Academic Websites Versus Faculty Web Pages](#).*
- *The Academic Designer/8 Benefits of a Personal Academic Website to Inspire You.*

If you're a junior scholar, having your own personal webpage is even more important.

- If (when) you move institutions, you'll lose your website.
- Even if you can export the contents of a WordPress site, there's no guarantee it will seamlessly integrate with another university's implementation.
- Even worse, you'll lose your search engine ranking since you'll be starting over from square one with a new URL.

Even if you stay at the same institution for the rest of your career, you're at the mercy of IT & your site may be taken down by a change to the hosting platform at some point in the future."

"There are plenty of guides out there on how to create a personal website using tools like WordPress, [Wix](#), or [Google Sites](#)<sup>3</sup>. The free versions of these tools often come with ads, or at the least a message telling you which tool was used to create the website." Rob Williams "use a [static site generator](#) that produces HTML from easy to edit [Markdown](#) files. Because the resulting site is static (it's just a collection of files with no interactivity where users can, e.g., fill out & submit forms) it can be hosted for free with [GitHub Pages](#). [Steven Miller](#) has a nice rundown on all of the [advantages](#) of this approach."

"This guide is intended for someone with a basic level of coding experience & comfort with Markdown files." "There are other guides to using static site generators to make academic websites, but they all assume a very high level of experience with the required tools & the ability to conduct extensive troubleshooting on your own."

"A brief aside on Git-speak: these periodic indented blocks will explain the terminology that git uses to help you understand what each Git command actually does."

"2 of the most popular programs available for building static sites from Markdown files are [Jekyll](#) & [Hugo](#)." "There are plenty of differences under the hood, but the most important one for building an academic website is that Hugo integrates nicely with the [blogdown](#) R package, letting you write your website entirely in R."

"Git-speak aside: the basic unit of GitHub is the repository. Repositories are just folders (directories, if you want to be pedantic), but Git keeps a record of the files in the folders. We'll start by making a repository on GitHub & then later download that repository to our computer. In both cases, it's just a folder. The magic of [Git](#) is that we can link the 2 so that changes you make in your local repository (the one on your computer) will sync with the remote repository (the one on GitHub). When people (myself included) get lazy, they'll often shorten repository to 'repo'."

"[Git](#) is a version control system designed to let teams of programmers collaborate on projects seamlessly. For us, it's just going to be the way that we upload files for our webpage to GitHub."

"Git-speak aside: cloning a repository means creating a local repository (folder) on your computer that's connected to the remote repository (on GitHub). Cloning differs from downloading in that you are setting up a connection between the 2 folders so you can keep changes you make locally synced up with the remote repository (which is where GitHub will build your website from)."

The `git fork` & `git clone` steps are straightforward.

## 1.2 Previewing Website

Once we upload our modified files to GitHub & tell GitHub to turn them into a website, they're out there on the Internet for everyone to see. "There's no need to broadcast all of those mistakes to the world, & we can avoid this very easily by previewing our website locally. What this means is building the site from the various `.md` files, rendering it to HTML, & then viewing it. We can do all of that on our computer without ever having to put it online."

"To preview your website locally, you'll need to install Jekyll on your computer. The easiest way to do this is with [Bundler](#). Bundler is a package manager for Ruby, which is the programming language that Jekyll is written in. This means that we need a full Ruby development environment to get Jekyll working to run our website locally."

To install Bundler, run:

---

<sup>3</sup>NQBH: I used Google Sites already. It is too slow to edit & publish. So I will not use Google Site anymore.

```

hong@hong-Katana-GF76-11UC:~/1_NQBH/NQBH.github.io$ sudo gem install bundler
Fetching bundler-2.3.12.gem
Successfully installed bundler-2.3.12
Parsing documentation for bundler-2.3.12
Installing ri documentation for bundler-2.3.12
Done installing documentation for bundler after 0 seconds
1 gem installed

```

“Next, we need to install any packages (called ‘gems’ in Ruby) that Jekyll depends on. This is where Bundler shines by taking care of this whole process for us; it reads the `Gemfile` included with the source code & install all required gems:

```
$ bundle install
```

If you want to see what’s been installed, run `gem list` before & after `bundle install`. If everything worked correctly, you can now launch your website! What we’re going to do is start a webserver on your computer, which will let you access your website locally without having to put it on the Internet. We do this with

```
bundle exec jekyll serve
```

The `bundle exec` command is just a prefix that lets Ruby access all of the gems specified in the `gemfile`. The `jekyll serve` command builds your website & starts a webserver so that you can view it locally. To access your website, open a browser & go to `127.0.0.1:4000` or `localhost:4000`.”

“This is a special version of your site that’s only accessible from your computer; no one else can see it! So this is the perfect place to play around, experiment, & see how to make your site do what you want it to. This process is surprisingly easy. Make a change to a file, e.g., editing `_pages/about.md` to introduce yourself, & save the file. That’s all you have to do; Jekyll will notice the change to the file & automatically rebuild the site. All that’s left to do is refresh your browser so you can see the changes!

Once you’ve made a couple changes to see how it works, you might want to turn off the webserver & make lots of changes, then check out your handiwork. Or maybe you’re just done working on your website for now. Either way, it’s time to shut down the webserver. To do so, you can just close the terminal window, but you’ll get a warning like this

```
Do you want to terminate running processes on this window?
```

```
Closing this window will terminate the running processes: jekyll serve --incremental, fsevent_watch
```

To save yourself some time & do this faster, simply press `CTRL + C`.”

Explicitly on my laptop:

```

hong@hong-Katana-GF76-11UC:~/1_NQBH/NQBH.github.io$ sudo bundle install
Don't run Bundler as root. Bundler can ask for sudo if it is needed, and
installing your bundle as root will break this application for all non-root
users on this machine.
Fetching gem metadata from https://rubygems.org/.....
Resolving dependencies....
Using concurrent-ruby 1.1.10
Using minitest 5.15.0
Using faraday-httpclient 1.0.1
Using thread_safe 0.3.6
Using zeitwerk 2.5.4
Using faraday-net_http_persistent 1.2.0
Using faraday-patron 1.0.0
Using faraday-rack 1.0.0
Using faraday-retry 1.0.3
Using ruby2_keywords 0.0.5
Using forwardable-extended 2.6.0
Using coffee-script-source 1.11.1
Using rb-fsevent 0.11.1
Using faraday-em_http 1.0.0
Using liquid 4.0.3
Using mercenary 0.3.6
Using rouge 3.26.0
Using safe_yaml 1.0.5
Fetching racc 1.6.0

```

```

Using faraday-em_synchrony 1.0.0
Using public_suffix 4.0.7
Using rubyzip 2.3.2
Using jekyll-swiss 1.0.0
Using unicode-display_width 1.8.0
Fetching unf_ext 0.0.8.1
Fetching commonmarker 0.23.4
Using colorator 1.1.0
Using faraday-excon 1.1.0
Fetching http_parser.rb 0.8.0
Using gemoji 3.0.1
Fetching ffi 1.15.5
Using jekyll-paginate 1.1.0
Using bundler 2.3.12
Fetching eventmachine 1.2.7
Using multipart-post 2.1.1
Using execjs 2.8.1
Using i18n 0.9.5
Using coffee-script 2.4.1
Using tzinfo 1.2.9
Using jekyll-coffeescript 1.1.1
Using rexml 3.2.5
Using addressable 2.8.0
Using kramdown 2.3.2
Using faraday-multipart 1.0.3
Using kramdown-parser-gfm 1.1.0
Using faraday-net_http 1.0.1
Using activesupport 6.0.4.7
Using faraday 1.10.0
Using pathutil 0.16.2
Using terminal-table 1.8.0
Using sawyer 0.8.2
Using octokit 4.22.0
Using jekyll-gist 1.5.0
Installing racc 1.6.0 with native extensions
Installing commonmarker 0.23.4 with native extensions
Installing http_parser.rb 0.8.0 with native extensions
Installing eventmachine 1.2.7 with native extensions
Installing unf_ext 0.0.8.1 with native extensions
Installing ffi 1.15.5 with native extensions
Gem::Ext::BuildError: ERROR: Failed to build gem native extension.

```

```

current directory:
/var/lib/gems/2.7.0/gems/commonmarker-0.23.4/ext/commonmarker
/usr/bin/ruby2.7 -I /usr/lib/ruby/vendor_ruby -r
./siteconf20220421-57640-1cwfd0z.rb extconf.rb
mkmf.rb can't find header files for ruby at /usr/lib/ruby/include/ruby.h

```

You might have to install separate package for the ruby development environment, ruby-dev or ruby-devel for example.

extconf failed, exit code 1

```

Gem files will remain installed in /var/lib/gems/2.7.0/gems/commonmarker-0.23.4
for inspection.
Results logged to
/var/lib/gems/2.7.0/extensions/x86_64-linux/2.7.0/commonmarker-0.23.4/gem_make.out

```

```
/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:91:in 'run'
```

```

/usr/lib/ruby/vendor_ruby/rubygems/ext/ext_conf_builder.rb:48:in 'block in
build'
  /usr/lib/ruby/2.7.0/tempfile.rb:291:in 'open'
  /usr/lib/ruby/vendor_ruby/rubygems/ext/ext_conf_builder.rb:28:in 'build'
  /usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:157:in 'build_extension'
/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:191:in 'block in
build_extensions'
  /usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:188:in 'each'
  /usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:188:in 'build_extensions'
  /usr/lib/ruby/vendor_ruby/rubygems/installer.rb:821:in 'build_extensions'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/rubygems_gem_installer.rb:71:in
'build_extensions'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/rubygems_gem_installer.rb:28:in
'install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/source/rubygems.rb:204:in
'install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/gem_installer.rb:54:in
'install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/gem_installer.rb:16:in
'install_from_spec'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/parallel_installer.rb:186:in
'do_install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/parallel_installer.rb:177:in
'block in worker_pool'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:62:in
'apply_func'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:57:in 'block in
process_queue'
  /var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:54:in 'loop'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:54:in
'process_queue'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:91:in 'block (2
levels) in create_threads'

```

An error occurred while installing commonmarker (0.23.4), and Bundler cannot continue.

In Gemfile:

```

github-pages was resolved to 226, which depends on
  jekyll-commonmark-ghpages was resolved to 0.2.0, which depends on
    jekyll-commonmark was resolved to 1.4.0, which depends on
      commonmarker

```

Gem::Ext::BuildError: ERROR: Failed to build gem native extension.

```

current directory: /var/lib/gems/2.7.0/gems/unf_ext-0.0.8.1/ext/unf_ext
/usr/bin/ruby2.7 -I /usr/lib/ruby/vendor_ruby -r
./siteconf20220421-57640-ehba7a.rb extconf.rb
mkmf.rb can't find header files for ruby at /usr/lib/ruby/include/ruby.h

```

You might have to install separate package for the ruby development environment, ruby-dev or ruby-devel for example.

extconf failed, exit code 1

Gem files will remain installed in /var/lib/gems/2.7.0/gems/unf\_ext-0.0.8.1 for inspection.  
Results logged to

```

/var/lib/gems/2.7.0/extensions/x86_64-linux/2.7.0/unf_ext-0.0.8.1/gem_make.out

/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:91:in 'run'
/usr/lib/ruby/vendor_ruby/rubygems/ext/ext_conf_builder.rb:48:in 'block in
build'
/usr/lib/ruby/2.7.0/tempfile.rb:291:in 'open'
/usr/lib/ruby/vendor_ruby/rubygems/ext/ext_conf_builder.rb:28:in 'build'
/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:157:in 'build_extension'
/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:191:in 'block in
build_extensions'
/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:188:in 'each'
/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:188:in 'build_extensions'
/usr/lib/ruby/vendor_ruby/rubygems/installer.rb:821:in 'build_extensions'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/rubygems_gem_installer.rb:71:in
'build_extensions'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/rubygems_gem_installer.rb:28:in
'install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/source/rubygems.rb:204:in
'install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/gem_installer.rb:54:in
'install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/gem_installer.rb:16:in
'install_from_spec'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/parallel_installer.rb:186:in
'do_install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/parallel_installer.rb:177:in
'block in worker_pool'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:62:in
'apply_func'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:57:in 'block in
process_queue'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:54:in 'loop'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:54:in
'process_queue'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:91:in 'block (2
levels) in create_threads'

```

An error occurred while installing unf\_ext (0.0.8.1), and Bundler cannot continue.

In Gemfile:

```

github-pages was resolved to 226, which depends on
github-pages-health-check was resolved to 1.17.9, which depends on
dnstruby was resolved to 1.61.9, which depends on
simpleidn was resolved to 0.2.1, which depends on
unf was resolved to 0.1.4, which depends on
unf_ext

```

Gem::Ext::BuildError: ERROR: Failed to build gem native extension.

current directory:

```

/var/lib/gems/2.7.0/gems/http_parser.rb-0.8.0/ext/ruby_http_parser
/usr/bin/ruby2.7 -I /usr/lib/ruby/vendor_ruby -r
./siteconf20220421-57640-zl5ph2.rb extconf.rb
mkmf.rb can't find header files for ruby at /usr/lib/ruby/include/ruby.h

```

You might have to install separate package for the ruby development environment, ruby-dev or ruby-devel for example.

extconf failed, exit code 1

Gem files will remain installed in /var/lib/gems/2.7.0/gems/http\_parser.rb-0.8.0 for inspection.  
Results logged to  
/var/lib/gems/2.7.0/extensions/x86\_64-linux/2.7.0/http\_parser.rb-0.8.0/gem\_make.out

```

/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:91:in 'run'
/usr/lib/ruby/vendor_ruby/rubygems/ext/ext_conf_builder.rb:48:in 'block in build'
/usr/lib/ruby/2.7.0/tempfile.rb:291:in 'open'
/usr/lib/ruby/vendor_ruby/rubygems/ext/ext_conf_builder.rb:28:in 'build'
/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:157:in 'build_extension'
/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:191:in 'block in build_extensions'
/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:188:in 'each'
/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:188:in 'build_extensions'
/usr/lib/ruby/vendor_ruby/rubygems/installer.rb:821:in 'build_extensions'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/rubygems_gem_installer.rb:71:in 'build_extensions'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/rubygems_gem_installer.rb:28:in 'install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/source/rubygems.rb:204:in 'install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/gem_installer.rb:54:in 'install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/gem_installer.rb:16:in 'install_from_spec'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/parallel_installer.rb:186:in 'do_install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/parallel_installer.rb:177:in 'block in worker_pool'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:62:in 'apply_func'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:57:in 'block in process_queue'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:54:in 'loop'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:54:in 'process_queue'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:91:in 'block (2 levels) in create_threads'

```

An error occurred while installing http\_parser.rb (0.8.0), and Bundler cannot continue.

In Gemfile:

```

github-pages was resolved to 226, which depends on
  jekyll-avatar was resolved to 0.7.0, which depends on
    jekyll was resolved to 3.9.2, which depends on
      em-websocket was resolved to 0.5.3, which depends on
        http_parser.rb

```

Gem::Ext::BuildError: ERROR: Failed to build gem native extension.

```

current directory: /var/lib/gems/2.7.0/gems/racc-1.6.0/ext/racc/cparse
/usr/bin/ruby2.7 -I /usr/lib/ruby/vendor_ruby -r
./siteconf20220421-57640-rk4k26.rb extconf.rb

```

mkmf.rb can't find header files for ruby at /usr/lib/ruby/include/ruby.h

You might have to install separate package for the ruby development environment, ruby-dev or ruby-devel for example.

extconf failed, exit code 1

Gem files will remain installed in /var/lib/gems/2.7.0/gems/racc-1.6.0 for inspection.

Results logged to

/var/lib/gems/2.7.0/extensions/x86\_64-linux/2.7.0/racc-1.6.0/gem\_make.out

```

/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:91:in 'run'
/usr/lib/ruby/vendor_ruby/rubygems/ext/ext_conf_builder.rb:48:in 'block in
build'
/usr/lib/ruby/2.7.0/tempfile.rb:291:in 'open'
/usr/lib/ruby/vendor_ruby/rubygems/ext/ext_conf_builder.rb:28:in 'build'
/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:157:in 'build_extension'
/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:191:in 'block in
build_extensions'
/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:188:in 'each'
/usr/lib/ruby/vendor_ruby/rubygems/ext/builder.rb:188:in 'build_extensions'
/usr/lib/ruby/vendor_ruby/rubygems/installer.rb:821:in 'build_extensions'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/rubygems_gem_installer.rb:71:in
'build_extensions'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/rubygems_gem_installer.rb:28:in
'install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/source/rubygems.rb:204:in
'install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/gem_installer.rb:54:in
'install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/gem_installer.rb:16:in
'install_from_spec'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/parallel_installer.rb:186:in
'do_install'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/installer/parallel_installer.rb:177:in
'block in worker_pool'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:62:in
'apply_func'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:57:in 'block in
process_queue'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:54:in 'loop'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:54:in
'process_queue'
/var/lib/gems/2.7.0/gems/bundler-2.3.12/lib/bundler/worker.rb:91:in 'block (2
levels) in create_threads'

```

An error occurred while installing racc (1.6.0), and Bundler cannot continue.

In Gemfile:

```

github-pages was resolved to 226, which depends on
  jekyll-mentions was resolved to 1.6.0, which depends on
    html-pipeline was resolved to 2.14.1, which depends on
      nokogiri was resolved to 1.13.4, which depends on
        racc

```

To fix this error, I followed the answer of the question [Stack Overflow/An error occurred while installing racc \(1.6.0\), and Bundler cannot continue:](#)



1. Update ruby to [the latest version 2.7.0p0](#) using:

```
sudo apt-get install ruby-full build-essential
```

2. Rerun `bundle install` (no need sudo).

Explicitly,

```
hong@hong-Katana-GF76-11UC:~/1_NQBH/NQBH.github.io$ sudo apt-get install ruby-full build-essential
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ri ruby-dev ruby2.7-dev ruby2.7-doc
The following NEW packages will be installed:
  build-essential ri ruby-dev ruby-full ruby2.7-dev ruby2.7-doc
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 2.421 kB of archives.
After this operation, 24,5 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://vn.archive.ubuntu.com/ubuntu impish/main amd64 build-essential amd64 12.9ubuntu2 [4.678 B]
Get:2 http://vn.archive.ubuntu.com/ubuntu impish-updates/main amd64 ruby2.7-doc all 2.7.4-1ubuntu3.1 [2.216 kB]
Get:3 http://vn.archive.ubuntu.com/ubuntu impish/universe amd64 ri all 1:2.7+2build1 [4.418 B]
Get:4 http://vn.archive.ubuntu.com/ubuntu impish-updates/main amd64 ruby2.7-dev amd64 2.7.4-1ubuntu3.1 [189 kB]
Get:5 http://vn.archive.ubuntu.com/ubuntu impish/main amd64 ruby-dev amd64 1:2.7+2build1 [4.524 B]
Get:6 http://vn.archive.ubuntu.com/ubuntu impish/universe amd64 ruby-full all 1:2.7+2build1 [2.586 B]
Fetched 2.421 kB in 1s (4.291 kB/s)
Selecting previously unselected package build-essential.
(Reading database ... 285417 files and directories currently installed.)
Preparing to unpack .../0-build-essential_12.9ubuntu2_amd64.deb ...
Unpacking build-essential (12.9ubuntu2) ...
Selecting previously unselected package ruby2.7-doc.
Preparing to unpack .../1-ruby2.7-doc_2.7.4-1ubuntu3.1_all.deb ...
Unpacking ruby2.7-doc (2.7.4-1ubuntu3.1) ...
Selecting previously unselected package ri.
Preparing to unpack .../2-ri_1%3a2.7+2build1_all.deb ...
Unpacking ri (1:2.7+2build1) ...
Selecting previously unselected package ruby2.7-dev:amd64.
Preparing to unpack .../3-ruby2.7-dev_2.7.4-1ubuntu3.1_amd64.deb ...
Unpacking ruby2.7-dev:amd64 (2.7.4-1ubuntu3.1) ...
Selecting previously unselected package ruby-dev:amd64.
Preparing to unpack .../4-ruby-dev_1%3a2.7+2build1_amd64.deb ...
Unpacking ruby-dev:amd64 (1:2.7+2build1) ...
Selecting previously unselected package ruby-full.
Preparing to unpack .../5-ruby-full_1%3a2.7+2build1_all.deb ...
Unpacking ruby-full (1:2.7+2build1) ...
Setting up ruby2.7-doc (2.7.4-1ubuntu3.1) ...
Setting up ruby2.7-dev:amd64 (2.7.4-1ubuntu3.1) ...
Setting up build-essential (12.9ubuntu2) ...
Setting up ruby-dev:amd64 (1:2.7+2build1) ...
Setting up ri (1:2.7+2build1) ...
Setting up ruby-full (1:2.7+2build1) ...
```

then rerun `bundle install`:

```
hong@hong-Katana-GF76-11UC:~/1_NQBH/NQBH.github.io$ bundle install
Fetching gem metadata from https://rubygems.org/.....
Resolving dependencies....
Using concurrent-ruby 1.1.10
Using faraday-em_http 1.0.0
Using thread_safe 0.3.6
```

```
Using zeitwerk 2.5.4
Using public_suffix 4.0.7
Using bundler 2.3.12
Using faraday-excon 1.1.0
Using coffee-script-source 1.11.1
Using colorator 1.1.0
Using faraday-em_synchrony 1.0.0
Using minitest 5.15.0
Using execjs 2.8.1
Following files may not be writable, so sudo is needed:
  /usr/local/bin
  /var/lib/gems/2.7.0
  /var/lib/gems/2.7.0/build_info
  /var/lib/gems/2.7.0/cache
  /var/lib/gems/2.7.0/doc
  /var/lib/gems/2.7.0/extensions
  /var/lib/gems/2.7.0/gems
  /var/lib/gems/2.7.0/plugins
  /var/lib/gems/2.7.0/specifications
Using multipart-post 2.1.1
Using faraday-rack 1.0.0
Using faraday-retry 1.0.3
Using rb-fsevent 0.11.1
Using mercenary 0.3.6
Using rouge 3.26.0
Using forwardable-extended 2.6.0
Using gemoji 3.0.1
Using faraday-net_http_persistent 1.2.0
Using rubyzip 2.3.2
Using unicode-display_width 1.8.0
Using faraday-httpclient 1.0.1
Using rexml 3.2.5
Using safe_yaml 1.0.5
Fetching unf_ext 0.0.8.1
Fetching eventmachine 1.2.7
Fetching commonmarker 0.23.4
Using faraday-patron 1.0.0
Using jekyll-swiss 1.0.0
Using liquid 4.0.3
Using pathutil 0.16.2
Using i18n 0.9.5
Using tzinfo 1.2.9
Fetching ffi 1.15.5
Using activesupport 6.0.4.7
Using faraday-net_http 1.0.1
Using faraday-multipart 1.0.3
Using ruby2_keywords 0.0.5
Using jekyll-paginate 1.1.0
Using faraday 1.10.0
Fetching racc 1.6.0
Using terminal-table 1.8.0
Using addressable 2.8.0
Using kramdown 2.3.2
Using sawyer 0.8.2
Fetching http_parser.rb 0.8.0
Using octokit 4.22.0
Using coffee-script 2.4.1
Using kramdown-parser-gfm 1.1.0
Using jekyll-coffeescript 1.1.1
```

Using jekyll-gist 1.5.0

Your user account isn't allowed to install to the system RubyGems.

You can cancel this installation and run:

```
bundle config set --local path 'vendor/bundle'
bundle install
```

to install the gems into ./vendor/bundle/, or you can enter your password and install the bundled gems to RubyGems using sudo.

Password:

```
Installing commonmarker 0.23.4 with native extensions
Installing racc 1.6.0 with native extensions
Installing http_parser.rb 0.8.0 with native extensions
Installing eventmachine 1.2.7 with native extensions
Installing unf_ext 0.0.8.1 with native extensions
Installing ffi 1.15.5 with native extensions
Fetching nokogiri 1.13.4 (x86_64-linux)
Fetching unf 0.1.4
Installing nokogiri 1.13.4 (x86_64-linux)
Fetching html-pipeline 2.14.1
Installing html-pipeline 2.14.1
Installing unf 0.1.4
Fetching simpleidn 0.2.1
Installing simpleidn 0.2.1
Fetching dnsruby 1.61.9
Installing dnsruby 1.61.9
Fetching jekyll-commonmark 1.4.0
Installing jekyll-commonmark 1.4.0
Fetching em-websocket 0.5.3
Installing em-websocket 0.5.3
Fetching ethon 0.15.0
Fetching rb-inotify 0.10.1
Installing rb-inotify 0.10.1
Installing ethon 0.15.0
Fetching sass-listen 4.0.0
Fetching listen 3.7.1
Fetching typhoeus 1.4.0
Installing listen 3.7.1
Installing sass-listen 4.0.0
Installing typhoeus 1.4.0
Fetching sass 3.7.4
Fetching jekyll-watch 2.2.1
Fetching github-pages-health-check 1.17.9
Installing jekyll-watch 2.2.1
Installing github-pages-health-check 1.17.9
Installing sass 3.7.4
Fetching jekyll-sass-converter 1.5.2
Installing jekyll-sass-converter 1.5.2
Fetching jekyll 3.9.2
Installing jekyll 3.9.2
Fetching jekyll-avatar 0.7.0
Fetching jekyll-mentions 1.6.0
Fetching jekyll-optional-front-matter 0.3.2
Fetching jekyll-readme-index 0.3.0
Fetching jemoji 0.12.0
Fetching jekyll-remote-theme 0.4.3
```

```
Fetching jekyll-seo-tag 2.8.0
Fetching jekyll-sitemap 1.4.0
Fetching jekyll-default-layout 0.1.4
Fetching jekyll-feed 0.15.1
Fetching jekyll-github-metadata 2.13.0
Fetching jekyll-include-cache 0.2.1
Fetching jekyll-titles-from-headings 0.5.3
Fetching jekyll-commonmark-ghpages 0.2.0
Fetching jekyll-relative-links 0.6.1
Fetching jekyll-redirect-from 0.16.0
Installing jekyll-optional-front-matter 0.3.2
Installing jekyll-readme-index 0.3.0
Installing jekyll-remote-theme 0.4.3
Installing jekyll-default-layout 0.1.4
Installing jemoji 0.12.0
Installing jekyll-relative-links 0.6.1
Installing jekyll-mentions 1.6.0
Installing jekyll-include-cache 0.2.1
Installing jekyll-redirect-from 0.16.0
Installing jekyll-commonmark-ghpages 0.2.0
Installing jekyll-titles-from-headings 0.5.3
Installing jekyll-avatar 0.7.0
Installing jekyll-feed 0.15.1
Installing jekyll-github-metadata 2.13.0
Installing jekyll-seo-tag 2.8.0
Installing jekyll-sitemap 1.4.0
Fetching hawkins 2.0.5
Fetching jekyll-theme-midnight 0.2.0
Fetching jekyll-theme-tactile 0.2.0
Fetching jekyll-theme-dinky 0.2.0
Fetching jekyll-theme-hacker 0.2.0
Fetching minima 2.5.1
Fetching jekyll-theme-leap-day 0.2.0
Fetching jekyll-theme-merlot 0.2.0
Fetching jekyll-theme-minimal 0.2.0
Fetching jekyll-theme-primer 0.6.0
Fetching jekyll-theme-slate 0.2.0
Fetching jekyll-theme-architect 0.2.0
Fetching jekyll-theme-cayman 0.2.0
Fetching jekyll-theme-modernist 0.2.0
Fetching jekyll-theme-time-machine 0.2.0
Installing jekyll-theme-hacker 0.2.0
Installing minima 2.5.1
Installing jekyll-theme-time-machine 0.2.0
Installing jekyll-theme-cayman 0.2.0
Installing jekyll-theme-modernist 0.2.0
Installing jekyll-theme-dinky 0.2.0
Installing jekyll-theme-primer 0.6.0
Installing jekyll-theme-tactile 0.2.0
Installing jekyll-theme-architect 0.2.0
Installing jekyll-theme-minimal 0.2.0
Installing hawkins 2.0.5
Installing jekyll-theme-slate 0.2.0
Installing jekyll-theme-merlot 0.2.0
Installing jekyll-theme-leap-day 0.2.0
Installing jekyll-theme-midnight 0.2.0
Fetching github-pages 226
Installing github-pages 226
Bundle complete! 4 Gemfile dependencies, 102 gems now installed.
```

Use 'bundle info [gemname]' to see where a bundled gem is installed.

Post-install message from dnsruby:

Installing dnsruby...

For issues and source code: <https://github.com/alexdalitz/dnsruby>

For general discussion (please tell us how you use dnsruby): <https://groups.google.com/forum/#!forum/dnsruby>  
 Post-install message from sass:

Ruby Sass has reached end-of-life and should no longer be used.

\* If you use Sass as a command-line tool, we recommend using Dart Sass, the new primary implementation: <https://sass-lang.com/install>

\* If you use Sass as a plug-in for a Ruby web framework, we recommend using the sassc gem: <https://github.com/sass/sassc-ruby#readme>

\* For more details, please refer to the Sass blog:  
<https://sass-lang.com/blog/posts/7828841>

Post-install message from html-pipeline:

-----  
 Thank you for installing html-pipeline!  
 You must bundle Filter gem dependencies.  
 See html-pipeline README.md for more details.  
<https://github.com/jch/html-pipeline#dependencies>  
 -----

"If you want to see what's been installed, run `gem list` before and after `bundle install`."

hong@hong-Katana-GF76-11UC:~/1\_NQBH/NQBH.github.io\$ gem list

\*\*\* LOCAL GEMS \*\*\*

```
activesupport (6.0.4.7)
addressable (2.8.0)
benchmark (default: 0.1.0)
bigdecimal (default: 2.0.0)
bundler (2.3.12, default: 2.1.4)
cgi (default: 0.1.0)
coffee-script (2.4.1)
coffee-script-source (1.11.1)
colorator (1.1.0)
commonmarker (0.23.4)
concurrent-ruby (1.1.10)
csv (default: 3.1.2)
date (default: 3.0.0)
dbm (default: 1.1.0)
delegate (default: 0.1.0)
did_you_mean (default: 1.4.0)
dnsruby (1.61.9)
em-websocket (0.5.3)
etc (default: 1.1.0)
ethon (0.15.0)
eventmachine (1.2.7)
execjs (2.8.1)
faraday (1.10.0)
faraday-em_http (1.0.0)
faraday-em_synchrony (1.0.0)
faraday-excon (1.1.0)
faraday-httpclient (1.0.1)
faraday-multipart (1.0.3)
```

faraday-net\_http (1.0.1)  
faraday-net\_http\_persistent (1.2.0)  
faraday-patron (1.0.0)  
faraday-rack (1.0.0)  
faraday-retry (1.0.3)  
fcntl (default: 1.0.0)  
ffi (1.15.5)  
fiddle (default: 1.0.0)  
fileutils (default: 1.4.1)  
forwardable (default: 1.3.1)  
forwardable-extended (2.6.0)  
gdbm (default: 2.1.0)  
gemoji (3.0.1)  
getoptlong (default: 0.1.0)  
github-pages (226)  
github-pages-health-check (1.17.9)  
hawkins (2.0.5)  
html-pipeline (2.14.1)  
http\_parser.rb (0.8.0)  
i18n (0.9.5)  
io-console (default: 0.5.6)  
ipaddr (default: 1.2.2)  
irb (default: 1.2.6)  
jekyll (3.9.2)  
jekyll-avatar (0.7.0)  
jekyll-coffeescript (1.1.1)  
jekyll-commonmark (1.4.0)  
jekyll-commonmark-ghpages (0.2.0)  
jekyll-default-layout (0.1.4)  
jekyll-feed (0.15.1)  
jekyll-gist (1.5.0)  
jekyll-github-metadata (2.13.0)  
jekyll-include-cache (0.2.1)  
jekyll-mentions (1.6.0)  
jekyll-optional-front-matter (0.3.2)  
jekyll-paginate (1.1.0)  
jekyll-readme-index (0.3.0)  
jekyll-redirect-from (0.16.0)  
jekyll-relative-links (0.6.1)  
jekyll-remote-theme (0.4.3)  
jekyll-sass-converter (1.5.2)  
jekyll-seo-tag (2.8.0)  
jekyll-sitemap (1.4.0)  
jekyll-swiss (1.0.0)  
jekyll-theme-architect (0.2.0)  
jekyll-theme-cayman (0.2.0)  
jekyll-theme-dinky (0.2.0)  
jekyll-theme-hacker (0.2.0)  
jekyll-theme-leap-day (0.2.0)  
jekyll-theme-merlot (0.2.0)  
jekyll-theme-midnight (0.2.0)  
jekyll-theme-minimal (0.2.0)  
jekyll-theme-modernist (0.2.0)  
jekyll-theme-primer (0.6.0)  
jekyll-theme-slate (0.2.0)  
jekyll-theme-tactile (0.2.0)  
jekyll-theme-time-machine (0.2.0)  
jekyll-titles-from-headings (0.5.3)  
jekyll-watch (2.2.1)

jemoji (0.12.0)  
json (default: 2.3.0)  
kramdown (2.3.2)  
kramdown-parser-gfm (1.1.0)  
liquid (4.0.3)  
listen (3.7.1)  
logger (default: 1.4.2)  
matrix (default: 0.2.0)  
mercenary (0.3.6)  
minima (2.5.1)  
minitest (5.15.0, 5.13.0)  
multipart-post (2.1.1)  
mutex\_m (default: 0.1.0)  
net-pop (default: 0.1.0)  
net-smtp (default: 0.1.0)  
net-telnet (0.1.1)  
nokogiri (1.13.4 x86\_64-linux)  
observer (default: 0.1.0)  
octokit (4.22.0)  
open3 (default: 0.1.0)  
openssl (default: 2.1.2)  
ostruct (default: 0.2.0)  
pathutil (0.16.2)  
power\_assert (1.1.7)  
prime (default: 0.1.1)  
pstore (default: 0.1.0)  
psych (default: 3.1.0)  
public\_suffix (4.0.7)  
racc (1.6.0, default: 1.4.16)  
rake (13.0.3)  
rb-fsevent (0.11.1)  
rb-inotify (0.10.1)  
rdoc (default: 6.2.1.1)  
readline (default: 0.0.2)  
readline-ext (default: 0.1.0)  
reline (default: 0.1.5)  
rexml (3.2.5, default: 3.2.3.1)  
rouge (3.26.0)  
rss (default: 0.2.8)  
ruby2\_keywords (0.0.5)  
rubygems-update (3.2.5)  
rubyzip (2.3.2)  
safe\_yaml (1.0.5)  
sass (3.7.4)  
sass-listen (4.0.0)  
sawyer (0.8.2)  
sdbm (default: 1.0.0)  
simpleidn (0.2.1)  
singleton (default: 0.1.0)  
stringio (default: 0.1.0)  
strscan (default: 1.0.3)  
terminal-table (1.8.0)  
test-unit (3.3.9)  
thread\_safe (0.3.6)  
timeout (default: 0.1.0)  
tracer (default: 0.1.0)  
typhoeus (1.4.0)  
tzinfo (1.2.9)  
unf (0.1.4)

```
unf_ext (0.0.8.1)
unicode-display_width (1.8.0)
uri (default: 0.10.0)
webrick (default: 1.6.1)
xmlrpc (0.3.0)
yaml (default: 0.1.0)
zeitwerk (2.5.4)
zlib (default: 1.1.0)
```

### 1.3 Getting Online

“... made some changes from the template, checked them out locally, & you’re ready to share your website with the world. This is a 2 step process. 1st we need to upload all of our modified files to the GitHub repo we forked from the template. Then we need to configure GitHub Pages to build & deploy our website. Finally, if you want a custom domain name, we need to do some configuration outside of GitHub Pages to connect your domain name with your website.”

#### 1.3.1 Uploading changes to GitHub

“To upload your changes to GitHub, we 1st have to make Git locally aware of them. We do this by committing the changes, then pushing them to the repo on GitHub.

“Git-speak aside: Git stores file histories as a series of changes or differences. A batch of changes (which can include changes in 1 or more files) is called a *commit*. When you want to tell the remote repo (the one on GitHub) about changes you’ve made, you push a commit from the local repo to the remote one. Once you do this, GitHub looks at the differences & modifies the files in the remote repo.”

Before we can commit the changes, we need to stage them. This just involved telling Git what changes we want to commit. To make our lives easier, let’s check in on what changes we’ve made by `git status`. You should get results that look something similar to this:

On branch master

Your branch is up to date with 'origin/master'.

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: \_config.yml

Untracked files:

(use "git add <file>..." to include in what will be committed)

Gemfile.lock

no changes added to commit (use "git add" and/or "git commit -a")

We can ignore the 1st part for now. The 2nd part (**Changes not staged for commit**) will list any files that Git knows about that have changed. The 3rd part (**Untracked files**) includes files we haven’t told Git about, so as far as it’s concerned they don’t exist.” “If you want to verify the changes you made, you’ll want to *diff* the file. Do this with

```
git diff _config.yml
```

[...] Note that your output may or may not be color-coded depending on what type of system you’re on & your Git settings. Each line that begins with a + indicates an insertion & each line that starts with a - is a deletion.” The `git commit` & `git push` steps are straightforward. “Your repo on GitHub is the “remote” that your local git needs access to.”

## 2 NQBH.github.io

Final result: URL: [nqbh.github.io](https://nqbh.github.io).



## 2.1 Hit Counter

See, e.g., [StackOverflow/How to implement a basic page view counter for a GitHub-Pages-powered site?](#)

### 2.1.1 JavaScript in Plain English/Mehdi Aoussiad. How to Count Page Views with The Count API

An easy way to count your page views with JavaScript.

“1 of the useful functionalities that you may need to have on your website is a counter that counts the views of your web pages. The easiest way to do that is through the CountAPI which gives you this amazing functionality.” “... learn how to count our page views using the CountAPI in JavaScript.”

**The CountAPI.** “This API allows you to make numeric counters. It’s a very useful API to track the number of hits a page received. It also allows you to know the number of users that, e.g., clicked on a button. So it also counts events. The CountAPI provides a hit endpoint each time it is called, the counter will increase by 1. Check it out in the link below: <https://api.countapi.xyz/hit/namespace/key>. You should know that each counter is identified inside a *namespace* with a *key*. The *namespace* should be unique, so you need to replace it with your website domain.”

**Example 2.1.** “... you want to count & display your web page views, the CountAPI allows you to do that by using their NPM package, JSONP, XHR, or jQuery. In the example below, we will use JSONP. You can check the other ways on <https://countapi.xyz/>. The 1st thing we do is creating our HTML:

```
<body>

  <h1>This page got <span id="visits"></span> views.</h1>

</body>
```

In the head tag, you will have to add a script CDN for the CountAPI. It will be like the following:

```
<head>

<script async src="https://api.countapi.xyz/hit/mysite.com/visits?  callback=callbackName"></script>

</head>
```

You will have to replace the `mysite.com` with your own domain. Also, replace the `callbackName` with any callback name that we will create below in JavaScript.

Now we will go to JavaScript in order to create a callback that will get the number of visits & replace it inside the element that has an ID `visits`.

```
function callbackName(response) {
  document.getElementById('visits').innerText = response.value;
}
```

We will not call this function because it’s already called on the CDN above.” “As you can see, this is the easiest way to count the views that your web page received.”

“The CountAPI makes it easier to count page views & display them on the page. This API also has some more functionalities ...”.

## 2.2 CountAPI

“This API allows you to create simple numeric counters. IaaS, Integer as a Service. It goes down to:

- Create a counter & restrict its operations
- Reset the value of a counter
- Increment/decrement a counter.

All counters are accessible if you know the key & there are not private counters (yet?). Want to track the number of hits a page had? Sure. Want to know the number of users that clicked on the button “Feed Cow”? There you go.”

**TL;DR.** “Each counter is identified inside a **namespace** with a **key**. The namespace should be unique, so its recommend using your site’s domain. In each namespace you can generate all the counters you may need.

The **hit** endpoint provides increment by 1 counters directly. Each time its requested the counter will increase by 1:  
<https://api.countapi.xyz/hit/namespace/key>  $\Rightarrow$  200 { "value": 1234 }.

**Example 2.2.** “... want to display the number of pageviews a site received.

```
<div id="visits">...</div>
```

*Remeber to change mysite.com with your site’s domain.*

#### Using countapi-js.

```
import countapi from 'countapi-js';

countapi.visits().then((result) => {
  console.log(result.value);
});
```

#### Using JSONP.

```
<script>
function cb(response) {
  document.getElementById('visits').innerText = response.value;
}
</script>
<script async src="https://api.countapi.xyz/hit/mysite.com/visits?callback=cb"></script>
```

#### Using XHR.

```
var xhr = new XMLHttpRequest();
xhr.open("GET", "https://api.countapi.xyz/hit/mysite.com/visits");
xhr.responseType = "json";
xhr.onload = function() {
  document.getElementById('visits').innerText = this.response.value;
}
xhr.send();
```

#### Using jQuery.

```
$.getJSON("https://api.countapi.xyz/hit/mysite.com/visits", function(response) {
  $("#visits").text(response.value);
});
```

**Multiple pages.** “If you want to have a counter for each individual page you can replace **visits** with a unique identifier for each page, i.e., **index**, **contact**, **item-1234**. Check **the right format** a key must have.

Alternatively, you can use some reserved words that are replaced server-side.

E.g., if a request is made from <https://mysite.com/example/page>:

- **:HOST**: will be replaced with **mysite.com**
- **:PATHNAME**: will be replaced with **examplepage**.

**Remark 2.1.** *Reserved words are padded with dots if their length is < 3.*

So you could use something like:

```
https://api.countapi.xyz/hit/mysite.com/:PATHNAME:
```

Or even more generic (though not recommended):

```
https://api.countapi.xyz/hit/:HOST/:PATHNAME:
```

**Remark 2.2** (Important). *If you want to know the actual key used you can check the **X-Path** header.”*

**Counting events.** “You can use the API to count any kind of stuff, lets say:

```
<button onclick="clicked()">Press Me!</button>
<script>
function clicked() {
  var xhr = new XMLHttpRequest();
  xhr.open("GET", "https://api.countapi.xyz/hit/mysite.com/awesomeclick");
  xhr.responseType = "json";
  xhr.onload = function() {
    alert('This button has been clicked ${this.response.value} times!');
  }
  xhr.send();
}
</script>
```

**Libraries.** “The official javascript promise based wrapper is available in [countapi-js](#) (v1.0.2).

**Roadmap.** “If this API starts getting some traction, I have some ideas in mind:

- Being able to open a [SSE stream](#) & receive updates in real time avoiding polling
- Enable batch creating/updating
- Unique counting?
- Float numbers?
- Generate a secret key to update/reset keys

## FAQ.

1. “*Is the API free?* Completely free.
2. *Rate Limiting?* Key retrieving & updating has **no** limits whatsoever. Key **creation** is limited to 20/IP/s.
3. *Can I delete a key?* You can’t, just let the key expire. If you created a key with a wrong configuration you can always create another key.
4. *Will I blow up the CountAPI server?* CountAPI is using [Redis](#) as database, a very fast key-value solution.”

## API.

**Namespaces.** “Namespaces are meant to avoid name collisions. You may specify a namespace during the creation of a key. Its recommend use the domain of the application as namespace to avoid collision with other websites. If the namespace is not specified the key is assigned to the **default** namespace. If your key resides in the default namespace you don’t need to specify it.”

**Endpoint.** “All requests support [cross-origin resource sharing](#) (CORS) & SSL. You can use [JSONP](#) sending the callback parameter. JSONP requests will never fail, they will include the HTTP code in the response. Also a  $1 \times 1$  GIF image is supported sending `?img`. Base API path: <https://api.countapi.xyz/>. In the case of a server failure, the API will send:  $\Rightarrow$  500 { "error": "Error description" }. For more technical details, see [CountAPI/API](#).

## References

[Williams2020] Rob Williams. *Building an Academic Website*. Jun 30, 2020.