

API Documentation Browser <https://devdocs.io>

[#devdocs](#) [#documentation](#) [#documentation-tool](#) [#developer-tools](#) [#docs](#) [#offline](#) [#app](#) [#api-documentation](#) [#pwa](#)

2,831 commits

3 branches

0 releases

93 contributors

View license

Branch: master ▾

New pull request

Create new file

Upload files

Find File

Clone or download ▾

raisedadead Add a code of conduct (#1006) ... Latest commit b5bd218 on May 6

.github	Documentation maintenance	2 months ago
assets	2019	6 months ago
docs	Documentation maintenance	2 months ago
lib	Bump Ruby 2.6.0	6 months ago
public	Add icon source files	7 months ago
test	Simplify file scraper setup; scrape files in the "docs/[slug]" directory	7 months ago
views	Copy changes to reflect that DevDocs is now operated by freeCodeCamp	7 months ago
.dockerignore	Improved dockerfile, and added alpine version	2 years ago
.editorconfig	Documentation maintenance	2 months ago
.gitignore	Add maintainer's guide	7 months ago
.image_optim.yml	Disable optipng	2 years ago
.ruby-version	Bump Ruby 2.6.0	6 months ago
.slugignore	Add Heroku config files	last year
.travis.yml	Tell Travis to cache bundle	9 months ago
CODE_OF_CONDUCT.md	Add a code of conduct	2 months ago
COPYRIGHT	2019	6 months ago
Dockerfile	Bump Ruby 2.6.0	6 months ago
Dockerfile-alpine	Bump Ruby 2.6.0	6 months ago
Gemfile	Bump Ruby 2.6.0	6 months ago
Gemfile.lock	Bump Ruby 2.6.0	6 months ago
LICENSE	Going open source	6 years ago
PULL_REQUEST_TEMPLATE.md	Don't require an icon	5 months ago
Procfile	Add Heroku config files	last year
README.md	Documentation maintenance	2 months ago
Rakefile	Fix tests and docker build	8 months ago
Thorfile	Going open source	6 years ago
config.ru	Going open source	6 years ago
newrelic.yml	Add Heroku config files	last year

README.md

DevDocs — API Documentation Browser

build passing

DevDocs combines multiple developer documentations in a clean and organized web UI with instant search, offline support, mobile version, dark theme, keyboard shortcuts, and more.

DevDocs was created by [Thibaut Courouble](#) and is operated by [freeCodeCamp](#).

Keep track of development news:

- Join the contributor chat room on [Gitter](#)
- Watch the repository on [GitHub](#)
- Follow [@DevDocs](#) on Twitter

Table of Contents: [Quick Start](#) · [Vision](#) · [App](#) · [Scraper](#) · [Commands](#) · [Contributing](#) · [Documentation](#) · [Plugins and Extensions](#) · [License](#) · [Questions?](#)

Quick Start

Unless you wish to contribute to the project, we recommend using the hosted version at [devdocs.io](#). It's up-to-date and works offline out-of-the-box.

DevDocs is made of two pieces: a Ruby scraper that generates the documentation and metadata, and a JavaScript app powered by a small Sinatra app.

DevDocs requires Ruby 2.6.0, libcurl, and a JavaScript runtime supported by [ExecJS](#) (included in OS X and Windows; [Node.js](#) on Linux). Once you have these installed, run the following commands:

```
git clone https://github.com/freeCodeCamp/devdocs.git && cd devdocs
gem install bundler
bundle install
bundle exec thor docs:download --default
bundle exec rackup
```

Finally, point your browser at [localhost:9292](#) (the first request will take a few seconds to compile the assets). You're all set.

The `thor docs:download` command is used to download pre-generated documentations from DevDocs's servers (e.g. `thor docs:download html css`). You can see the list of available documentations and versions by running `thor docs:list`. To update all downloaded documentations, run `thor docs:download --installed`.

Note: there is currently no update mechanism other than `git pull origin master` to update the code and `thor docs:download --installed` to download the latest version of the docs. To stay informed about new releases, be sure to [watch](#) this repository.

Alternatively, DevDocs may be started as a Docker container:

```
# First, build the image
git clone https://github.com/freeCodeCamp/devdocs.git && cd devdocs
docker build -t thibaut/devdocs .

# Finally, start a DevDocs container (access http://localhost:9292)
docker run --name devdocs -d -p 9292:9292 thibaut/devdocs
```

Vision

DevDocs aims to make reading and searching reference documentation fast, easy and enjoyable.

The app's main goals are to: keep load times as short as possible; improve the quality, speed, and order of search results; maximize the use of caching and other performance optimizations; maintain a clean and readable user interface; be fully functional offline; support full keyboard navigation; reduce "context switch" by using a consistent typography and design across all documentations; reduce clutter by focusing on a specific category of content (API/reference) and indexing only the minimum useful to most developers.

Note: DevDocs is neither a programming guide nor a search engine. All our content is pulled from third-party sources and the project doesn't intend to compete with full-text search engines. Its backbone is metadata; each piece of content is identified by a unique, "obvious" and short string. Tutorials, guides and other content that don't meet this requirement are outside the scope of the project.

App

The web app is all client-side JavaScript, written in [CoffeeScript](#), and powered by a small [Sinatra/Sprockets](#) application. It relies on files generated by the [scraper](#).

Many of the code's design decisions were driven by the fact that the app uses XHR to load content directly into the main frame. This includes stripping the original documents of most of their HTML markup (e.g. scripts and stylesheets) to avoid polluting the main frame, and prefixing all CSS class names with an underscore to prevent conflicts.

Another driving factor is performance and the fact that everything happens in the browser. `applicationCache` (which comes with its own set of constraints) and `localStorage` are used to speed up the boot time, while memory consumption is kept in check by allowing the user to pick his/her own set of documentations. The search algorithm is kept simple because it needs to be fast even searching through 100,000 strings.

DevDocs being a developer tool, the browser requirements are high:

- Recent versions of Firefox, Chrome, or Opera
- Safari 9.1+
- Edge 16+
- iOS 10+

This allows the code to take advantage of the latest DOM and HTML5 APIs and make developing DevDocs a lot more fun!

Scraper

The scraper is responsible for generating the documentation and index files (metadata) used by the [app](#). It's written in Ruby under the `docs` module.

There are currently two kinds of scrapers: `UrlScraper` which downloads files via HTTP and `FileScraper` which reads them from the local filesystem. They both make copies of HTML documents, recursively following links that match a set of rules and applying all sorts of modifications along the way, in addition to building an index of the files and their metadata. Documents are parsed using [Nokogiri](#).

Modifications made to each document include:

- removing content such as the document structure (`<html>` , `<head>` , etc.), comments, empty nodes, etc.
- fixing links (e.g. to remove duplicates)
- replacing all external (not scraped) URLs with their fully qualified counterpart
- replacing all internal (scraped) URLs with their unqualified and relative counterpart
- adding content, such as a title and link to the original document
- ensuring correct syntax highlighting using [Prism](#)

These modifications are applied via a set of filters using the [HTML::Pipeline](#) library. Each scraper includes filters specific to itself, one of which is tasked with figuring out the pages' metadata.

The end result is a set of normalized HTML partials and two JSON files (index + offline data). Because the index files are loaded separately by the [app](#) following the user's preferences, the scraper also creates a JSON manifest file containing information about the documentations currently available on the system (such as their name, version, update date, etc.).

More information about [scrapers](#) and [filters](#) is available in the `docs` folder.

Available Commands

The command-line interface uses [Thor](#). To see all commands and options, run `thor list` from the project's root.

```
# Server
rackup          # Start the server (ctrl+c to stop)
rackup --help   # List server options

# Docs
thor docs:list   # List available documentations
thor docs:download # Download one or more documentations
thor docs:manifest # Create the manifest file used by the app
thor docs:generate # Generate/scrape a documentation
thor docs:page    # Generate/scrape a documentation page
thor docs:package # Package a documentation for use with docs:download
thor docs:clean   # Delete documentation packages

# Console
thor console     # Start a REPL
```

```
thor console:docs  # Start a REPL in the "Docs" module
Note: tests can be run quickly from within the console using the "test" command. Run "help test"
for usage instructions.

# Tests
thor test:all      # Run all tests
thor test:docs     # Run "Docs" tests
thor test:app      # Run "App" tests

# Assets
thor assets:compile # Compile assets (not required in development mode)
thor assets:clean   # Clean old assets
```

If multiple versions of Ruby are installed on your system, commands must be run through `bundle exec`.

Contributing

Contributions are welcome. Please read the [contributing guidelines](#).

Documentation

- [Adding documentations to DevDocs](#)
- [Scraper Reference](#)
- [Filter Reference](#)
- [Maintainers' Guide](#)

Plugins and Extensions

- [Chrome web app](#)
- [Ubuntu Touch app](#)
- [Sublime Text plugin](#)
- [Atom plugin](#)
- [Brackets extension](#)
- [Fluid](#) for turning DevDocs into a real OS X app
- [GTK shell / Vim integration](#)
- [Emacs lookup](#)
- [Alfred Workflow](#)
- [Vim search plugin with Devdocs in its defaults](#) Just set `let g:www_shortcut_engines = { 'devdocs': ['Devdocs', '<leader>dd'] }` to have a `:Devdocs` command and a `<leader>dd` mapping.
- [Visual Studio Code plugin](#) (1)
- [Visual Studio Code plugin](#) (2)
- [Desktop application](#)
- [Doc Browser](#) is a native Linux app that supports DevDocs docsets
- [GNOME Application](#) GTK3 application with search integrated in headerbar
- [macOS Application](#)
- [Android Application](#) is a fully working, advanced WebView with AppCache enabled

Copyright / License

Copyright 2013-2019 Thibaut Courouble and [other contributors](#)

This software is licensed under the terms of the Mozilla Public License v2.0. See the [COPYRIGHT](#) and [LICENSE](#) files.

Please do not use the name DevDocs to endorse or promote products derived from this software without the maintainers' permission, except as may be necessary to comply with the notice/attribution requirements.

We also wish that any documentation file generated using this software be attributed to DevDocs. Let's be fair to all contributors by giving credit where credit's due. Thanks!

Questions?

If you have any questions, please feel free to ask them on the contributor chat room on [Gitter](#).