

Jekyll AsciiDoc Quickstart

Table of Contents

Directions	1
1. Install Minimum Jekyll Requirements	1
2. Install Travis Gem	2
3. Fork this Repository and Clone	2
4. Enable Travis CI	2
5. Generate a GitHub Personal Access Token	2
6. Encrypt the GitHub Token for Travis CI	2
7. Verify the Configuration	3
Summary	3
Details	3
GitHub Pages and AsciiDoc	3
How We Work Around The Limitation	4
Help Get AsciiDoc Whitelisted for GitHub Pages	4
Repository Structure	4

The Jekyll AsciiDoc Quickstart project is a leg-up in starting your own website hosted on GitHub with content based in AsciiDoc. This project combines the power of AsciiDoc with a beautiful CSS framework and blog-ready template on top of GitHub's existing publishing infrastructure.

Directions

The goal of this procedure set is to configure a Travis CI job to listen for commits on the *master* branch, automatically run the Jekyll build, and push the generated content to the *gh-pages* branch.

1. Install Minimum Jekyll Requirements

You must install some software to execute commands in subsequent procedures. The Requirements on the [Jekyll Installation](#) page describe how to install both ruby and rubygems.

For yum-based package managers, the command to run is:

```
$ sudo yum install ruby rubygems
```

2. Install Travis Gem

When you install rubygems, you can use the gem internal package management system to install the Travis CI gem. This gem contains—among other things—a command-line tool for easily encrypting GitHub tokens.

Run the following command to install the Travis gem:

```
$ gem install travis
```

3. Fork this Repository and Clone

To create your own copy of this repository, start by clicking the fork button in the upper right corner of the GitHub page.

Next, open a command line window and make a clone of your new repository:

```
$ git clone https://github.com/YOUR-USERNAME/jekyll-asciidoc-quickstart
```

4. Enable Travis CI

Travis CI is configured initially through a browser.

To activate Travis CI for the Repository:

1. Open <https://travis-ci.org> and create an account.
2. Open your [profile page](#) on Travis.
3. Find the Jekyll repository, and turn on the switch.
4. Click on repository settings (next to the switch) and enable “Build only if .travis.yml is present.”

5. Generate a GitHub Personal Access Token

Once the repository is activated in Travis, you need a GitHub token to pass into the Travis keytool.

To generate a new personal access token on GitHub:

1. Open <https://github.com/settings/tokens/new>.
2. Select the scope *public_repository*, and add a terse description.
3. Confirm and save the settings.

6. Encrypt the GitHub Token for Travis CI

With the GitHub token created, you can now pass it to the Travis command-line tool, which adds

the encrypted value to a file in your repository.

To encrypt the token and add it to the `.travis.yml` file in your cloned repository:

1. Move into the same directory as `.travis.yml`.
2. Run the following command, replacing `<token>` with the GitHub token from the previous step.

```
$ travis encrypt GH_TOKEN=<token> --add env.global
```

3. Verify the script added the `secure` global environment variable to `.travis.yml`:

```
env:  
  global:  
    secure: [YOUR-ENCRYPTED-TOKEN]
```

4. Commit all changes, and push to GitHub.

```
$ git push
```

7. Verify the Configuration

To verify if you have configured the repository correctly, open <https://travis-ci.org> and verify that Travis starts, and subsequently finishes processing the job.

Travis should place the built site into the `gh-pages` branch upon completion.

Summary

If you can load the `[username].github.io/jekyll-asciidoc-quickstart` home page, you have successfully completed basic configuration.

Start writing blog posts and enjoy the AsciiDoc difference, regardless of what device you choose: computer, tablet, or mobile.

If you would like to work offline first, consider to check your `Gemfile` for the versions of your gem sources. As the case may be, your `jekyll serve` will fail. So make sure to check your gem sources with `gem list --local | grep "jekyll"`.

Details

GitHub Pages and AsciiDoc

GitHub Pages does not (yet) whitelist the jekyll-asciidoc plug-in, so you can not write `.adoc` posts

and have them instantly publish like Markdown posts do.

Unlike some "fork and write" repositories that exist for Markdown blogs, you need to initially configure this repository fork with a computer to publish using AsciiDoc.

How We Work Around The Limitation

For this repository, the [Travis CI](#) Continuous Integration (CI) server emulates GitHub Pages staging automation, and pushes your blog live upon committing any change to the repository.

After initially configuring the repository, you can use Git command-line on your computer, or even a Git client on your tablet or smartphone to write, commit, and automatically publish blog posts.

Help Get AsciiDoc Whitelisted for GitHub Pages

You can help change the lack of native AsciiDoc support by creating a support case through <http://github.com/support>.

Tell the GitHub team that you want the choice to write in AsciiDoc, and have it handled the same way Markdown is when pushed to your GitHub Page.

Your voice counts: make it heard!

Repository Structure

The repository requires the following structure to work correctly:

- **master**, for markup sources and configuration. This branch can be named anything you choose, however **master** is a general standard used in Jekyll blogs.
- **gh-pages**, for the generated static content produced by Travis CI. This branch is the `username.github.io` GitHub Pages domain, which is created automatically for you when the Travis CI job runs.