EzBlog Documentation

02-02-2018

Contents

This document will show you the basic usage of EzBlog. Found bugs or got something to complain? Tell me! ventor@ventora.net

1 Overview

EzBlog is a blogging software written in Bash. This way it is very light weight and can be used for serving a blog on a small server. EzBlog does not require a database, because every article is saved in a text file. Every article includes date, author and a title, which is read out by the "compiler" or "parser" that converts the markdown files into an usable HTML document. It is licensed under GPL3 and can be modified and shared, because it is free software.

1.1 Features

Here is a brief overview of what EzBlog **can** do for you:

- Serve a light weight blog
- Writing articles using the command line (for example via SSH)
- Using markdown syntax for writing articles
- Easy customization to the css stylesheet
- Many configuration options
- Generate an RSS feed

1.2 How it works

EzBlog consists of three main scripts. All of those scripts have a specific function.

- index.cgi
- viewarticle.cgi
- compile.sh

The following descriptions are very technical. Only people with a bit of "bash knowledge" will understand those terms.

1.2.1 index.cgi

This is the main file, that you will see when you load EzBlog. It includes the main logic for "drawing" the webpage. Basically, index.cgi starts a loop, looking for all files in the articles/ folder. If it finds an article, it will display the metadata of the article by using a few functions. Those functions use grep and cut (GNU utilities) for getting the required data. The only thing that those commands can see are the first 5 lines of the article.

If index.cgi finds a markdown (.md) file in the articles/ folder, it looks for a corresponding html file in the encoded/ folder. The html files are generated using the compile.sh script.

The script also includes a search function. The browser requests the following:

```
index.cgi?s=test
```

We use everything behind the ?s to make a quick grep command, which returns all the files, which include the search term. Now index.cgi is using the same loop to go through the search results and displaying the corresponding articles.

1.2.2 viewarticle.cgi

The viewarticle.cgi script needs an "GET" argument in the browser. It checks for the argument. For example the browser requests the following:

```
viewarticle.cgi?article=1
```

The script just reads out the corresponding data of the requested article.

1.2.3 compile.sh

compile.sh needs pandoc to run. It simply takes the markdown file in the articles folder and exports an html file to the encoded directory. As pandoc does not use the first 5 lines, it does work without any configuration. You only get the "raw" html file, which is good.

This way index.cgi can display the contents of the article by displaying the html file.

It is also planned that compile.sh gets a live mode feature. This enables you to run compile.sh in the background and the script is constantly looking for changes in the articles/ directory.

2 Installation

2.1 Requirements

EzBlog only requires a webserver and the "compiler/parser" pandoc.

- Bash
- A webserver, capable of serving Bash-CGI documents (I'm using apache2)
- pandoc

If you want to use the live mode of the compiler script, you need to download the inotify-tools from your repository.

Optional:

- inotifywait (Package inotify-tools)

2.2 First setup

Be sure to enable CGI on your webserver!!

Manuals can be found here:

- Configure it on apache2

2.2.1 Download & configure EzBlog

You can download EzBlog from the GitHub page, or by using git itself.

Here is an example for installing EzBlog using a Debian based system.

```
#Installing dependencies
sudo apt install pandoc inotify-tools

#Change to a public webserver directory
cd /var/www/html/blog/

#Cloning the repo
git clone https://github.com/flymia/EzBlog

#Move the files to the directory
mv /var/www/html/blog/EzBlog/* /var/www/html/blog

#Configure the blog using a text editor
nano /var/www/html/blog/config/settings
```

The settings file should be self explanatory. Just edit the file to your needs. For example, change the title and subtitle of your blog.

2.3 Your first article

It's time for your first article. Head over to the articles directory and make a new file called 1.md using your favorite text editor.

```
cd articles
vim 1.md #I'm using vim. You can use every other editor you wish.
```

Now comes the important stuff: It is essential to format the first lines correctly. These lines are read out by the index.cgi file and need to be correct! The default scheme of an article looks like this:

```
---
title: My cool blog post
author: Max
date: 22. January, 2018
---

This is the content area.
```

When you finished writing your article, save the file.

The next step is to compile/parse the article to HTML. You'll find compile.sh in the EzBlog root directory. Just run the command to compile the new article.

```
./compile.sh
```

You also can recompile every article, which is useful on the first setup.

```
./compile.sh --recompile
```

You can have a look at the encoded/ folder now. The compile script exported a HTML document now. The index.cgi page should now display the new entry in the browser.

2.4 Summary

1. Download EzBlog
2. Configure the settings file
3. Create a new article using a text editor in the articles directory
4. Compile the new article using compile.sh
5. View the webpage

3 Customization

It is also possible to customize the CSS style to your needs. Copy the default style sheet in the styles/ folder and start working on a new one!

4 Special Thanks

I want to thank everyone who worked on my previous project, SpotCurrent.

And a special thanks to Parckwart, who showed me the wonderful world of Bash cgi and helped me with a lot of problems.