FliKISS Documentation

Release 0.1

TROUVERIE Joachim

March 04, 2015

Contents

1 Install FliKISS **3**
1.1 With pip 3
1.2 Without pip 3

2 FliKISS config **5**

3 Run FliKISS **7**
3.1 Command 7
3.2 Run in background 7

4 Deploy FliKISS **9**
4.1 Launch on port 80 9
4.2 Reverse proxy 9

5 Use FliKISS **11**
5.1 General 11
5.2 Syntax 12

6 Aknowledgments **15**

FliKISS is a KISS wiki engine inspired by BlazeKISS a fork of Wikiss and developed with Flask.

Write your contents using the Markdown syntax with a nice editor without needing to install a database. Pages are stored in flat files on the disk.

CHAPTER 1

Install FliKISS

You need to have Python installed on your server to use FliKISS. The install script will download for you all the dependencies.

1.1 With pip

If you already have `pip` installed on your server you can use it to install FliKISS

```
$ pip install flikiss
```

It will download FliKISS from Pypi

1.2 Without pip

You can install `pip` easily using your package manager or install FliKISS manually following these steps :

1. Get the last archive of FliKISS
2. Extract it
3. Change path to extracted archive
4. Run `python setup.py install`

Note: Virtualenvs note Keep in mind that OS will otfen require you to prefix the above commands with *sudo* in order to install the app system-wide. It is recommended to create a virtual environment for FliKISS via virtualenv

CHAPTER 2

FliKISS config

By default FliKISS will look for a config file in your home `$HOME/.flikissrc`.

You can use an alternate config file using the command's `config` parameter (see the *Run section* for more details).

Settings are configured in the form of a Python module (a file) and need to respect the Python syntax. All the setting identifiers must be set in all-caps, otherwise they will not be processed.

Here is a list of settings you can overwrite :

Setting name(default value)	What does it do
PASSWORD (`password`)	Password to create or edit files
PAGES_DIR (`$FliKISS_path/pages`)	Absolute path to store your contents
MEDIA_DIR (`$FliKISS_path/media`)	Absolute path to store your media
NAMESPACES (`[]`)	You can use namespaces to categorize your pages. Namespaces are like new FliKISS instances, with their own menu and pages. While pages are similar to files, namespaces are similar to folders. Each namespace will be served on a sub-url using its name (for example a namespace named *wiki* will be served on *http://your-servername.com/wiki/*)

Here is an example of valid config file:

```
PASSWORD = 'A_password_very_hard_to_find'
PAGES_DIR = '/home/test/wiki/pages/'
MEDIA_DIR = '/home/test/wiki/media'
NAMESPACES = ['wiki','test']
```

Warning: FliKISS does not need a config file to be run, but you **should** create one to overwrite at least the default password(`password`).

CHAPTER 3

Run FliKISS

3.1 Command

FliKISS comes with the command `flikiss` to launch a CherryPy server to serve the application.

If you have installed it in a virtualenv you need to prefix it with your virtualenv path : `$PATH_TO_VIRTUALENV/bin/flikiss`

```
$ flikiss --ip 127.0.0.1 --port 8000 --url / --config /home/test/.flikissrc
```

-i, --ip
determine the host to serve the application (default=127.0.0.1)

-p, --port
determine the port to serve the application (default=8000)

-u, --url
determine the url to serve the application (default=/)

-c, --config
alternate config file

That's it, FliKISS now run on the port given in argument you can then access it at `http://ip:port/url/`

3.2 Run in background

FliKISS does not come with something build for this. You have several solutions.

3.2.1 Shell background process

To run a command in background you can run it in a shell background process with `nohup`

```
$ nohup flikiss [-options]
```

Or run it in a screen

```
$ screen
$ flikiss [-options]
```

Press `Ctrl` + `A` then `d`. Your session keep going on in background.

3.2.2 Supervisor

Supervisor is a program to manage processes, it can be easily installed using your package manager or `pip`.

Create a configuration file `/etc/supervisor/conf.d/flikiss.conf`

```
[program:flikiss]
command=flikiss [-options]
directory=/path-to-home/
environment=HOME='/path-to-home/'
autostart=true
autorestart=true
```

Then you need to enable your config file using as root

```
# supervisorctl update
# supervisorctl start your-app
```

CHAPTER 4

Deploy FliKISS

FliKISS is a Flask application which is served by the CherryPy WSGI container.

4.1 Launch on port 80

Note: On Unix systems ports below to 1024 are reserved to processes run by root

CherryPy stands on its own, but as an application server, it is often located in shared or complex environments.

If it is not your case you can deploy FliKISS launching it with the following parameters as a daemon

```
$ flikiss --ip 0.0.0.0 --port 80
```

Note: The ip set to `0.0.0.0` allows users to access your website from outside.

4.2 Reverse proxy

It is not uncommon to run CherryPy behind a reverse proxy if you already have a web server installed preventing you to use port 80.

First you need to run FliKISS as a *daemon*.

Here's for example a simple Nginx configuration which proxies to an application served on localhost at port 8000. You can easily adapt it to your needs

```
server {
    listen 80;
    server_name www.yourwebsite.com

    location </your-sub-url>/static/ {
        alias /path/to/flikiss/flikiss/static/;
    }

    location </your-sub-url>/ {
        proxy_pass http://127.0.0.1:8000;
        proxy_set_header Host $http_host;
    }
}
```

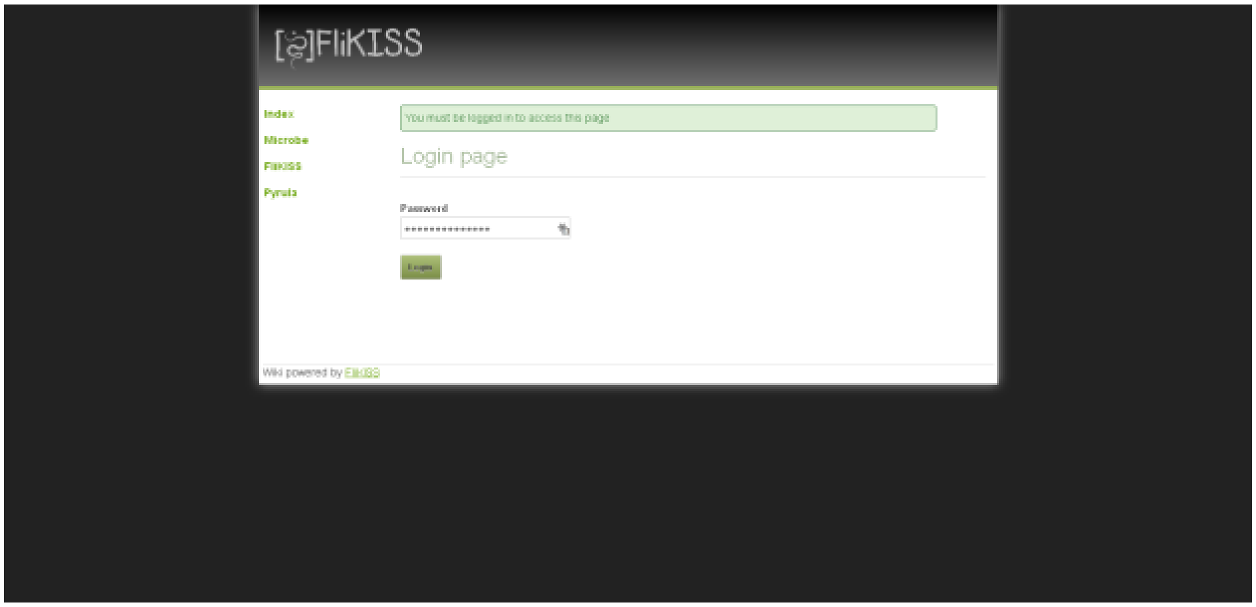
You can make some adjustments to get a better user experience

```
server {
    listen 80;
    server_name www.yourwebsite.com

    location </your-sub-url>/static/ {
            alias /path/to/flikiss/flikiss/static/;
            gzip  on;
            gzip_http_version 1.0;
            gzip_vary on;
            gzip_comp_level 6;
            gzip_proxied any;
            gzip_buffers 16 8k;
            gzip_disable ~@~\MSIE [1-6].(?!.*SV1)~@~];
            expires modified +90d;
    }

    location </your-sub-url>/ {
            proxy_pass 127.0.0.1:8000;
            proxy_set_header Host $http_host;
    }

}
```

Use FliKISS

5.1 General

Login



You must be logged in to access this page

Login page

Password

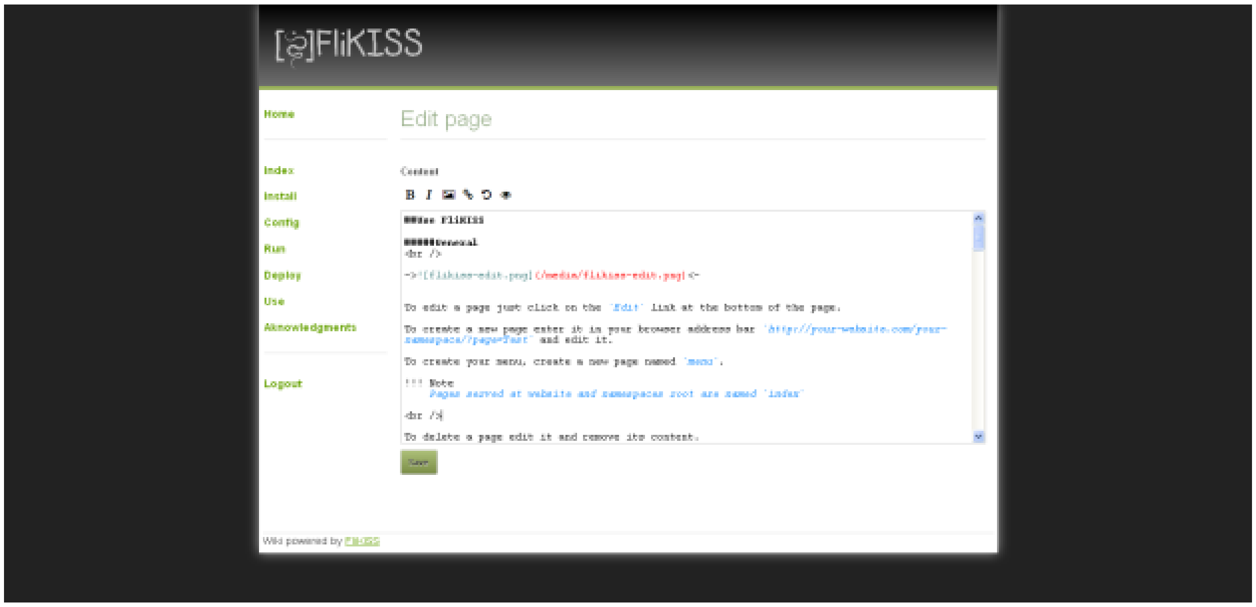
To create/edit or delete a page, you must be logged in the FliKISS app.

Enter the password you registered in your *config file*.

To log out, just click the link `Logout` at the bottom of your left menu.

Edit

To edit a page just click on the `Edit` link at the bottom of the page.



```
##Use FliKISS

#####General
<br />

->![flikiss-edit.png](/media/flikiss-edit.png)<-

To edit a page just click on the `Edit` link at the bottom of the page.

To create a new page enter it in your browser address bar `http://your-website.com/your-
namespace/?page=Test` and edit it.

To create your menu, create a new page named `menu`.

!!! Note
    Pages served at website and namespaces root are named `index`

<br />

To delete a page edit it and remove its content.
```

Once you have finish to edit a page just click on `Save` button to save your modifications.

Create

To create a new page enter its name in your browser address bar `http://your-website.com/your-namespace/?page=Test` and edit it.

To create a menu, create a new page called `menu`.

Note: Pages served at website and namespaces root are named `index`

Delete

To delete a page edit it and remove its content.

5.2 Syntax

FliKISS supports Markdown syntax with several extensions.

Table extension

You can easily create HTML tables with this extension using the sytax established in PHP Markdown extra.

Thus, the following text

```
My first header | My second header
--------------- | ----------------
Cell 1          | Cell 2
Cell 3          | Cell 4
```

will be rendered as

My first header	My second header
Cell 1	Cell 2
Cell 3	Cell 4

Codehilite extension

The CodeHilite extension follows the same syntax as regular Markdown code blocks, with one exception. The hiliter needs to know what language to use for the code block. It will use Pygments to highlight syntax. If the first line begins with three or more colons, the text following the colons identifies the language. The first line is removed from the code block before processing and line numbers are not used

```
:::python
def main(*args) :
    """
        Main function
    """
    if 'i' in 'this is a test' :
        print 'test'
```

will be rendered as:

```
def main(*args) :
    """
        Main function
    """
    if 'i' in 'this is a test' :
        print 'test'
```

Inter pages links

To make it easier to link content in your pages, FliKISS supports WikiLinks syntax.

It is possible to use a different label suffixing your link with a `|` and your label or to make links between namespaces, prefixing your link with your namespace and `:`.

The following text

```
[[Index]]
[[Index|Home]]
[[wiki:Index|My index]]
```

will be rendered as

```
<a href="/?page=Index">Index</a>
<a href="/?page=Index">Home</a>
<a href="/wiki/?page=Index">My Index</a>
```

Admonitions

The Admonition extension adds admonitions to Markdown documents. FliKISS comes with `Note` and `Warning` classes support.

The following text

```
!!! Admonition_class "A title"
    My text

!!! Note "A note"
    This is a note

!!! Warning "A warning"
    This is a warning
```

will be rendered as

```
<div class="admonition Admonition_class">
  <p class="admonition-title">A title</p>
  <p>My text</p>
```

```
</div>
<div class="admonition note">
  <p class="admonition-title">A note</p>
  <p>This is a note</p>
</div>
<div class="admonition warning">
  <p class="admonition-title">A warning</p>
  <p>This is a warning</p>
</div>
```

Content alignment

The Markdown syntax does not come with a functionnality to easily align your contents.

FliKISS comes with Mou syntax to center or right align your contents.

```
-> A center content <-
-> A right align content ->
```

will be rendered as

```
<div style="display:block;text-align:center;"> A center content </div>
<div style="display:block;text-align:right;"> A right align content </div>
```

Drad and drop

FliKISS editor supports HTML5 drag and drop API. If your browser supports it you can drop pictures directly in your editor to upload it.

You can also drop plain text files to fill your editor with its content.

CHAPTER 6

Aknowledgments

- BlazeKiss by Idleman
- Flask framework
- CherryPy
- CodeMirror
- Markdown editor inspired from lepture's
- Skeleton CSS framework
- Font awesome
- Upload by drag and drop by HTML5 demos

Index

Symbols

-c, –config
    command line option, 7
-i, –ip
    command line option, 7
-p, –port
    command line option, 7
-u, –url
    command line option, 7

C

command line option
    -c, –config, 7
    -i, –ip, 7
    -p, –port, 7
    -u, –url, 7