tailon Documentation

Release 1.4.2

Georgi Valkov

Jul 15, 2018

Contents

1 Screenshots **3**

2 Installation **7**

3 Quick start **9**
3.1 Tailon 9

4 Reverse proxy configuration **11**
4.1 Nginx 11

5 Security **13**

6 Development **15**

7 Similar Projects **17**

8 Attributions **19**

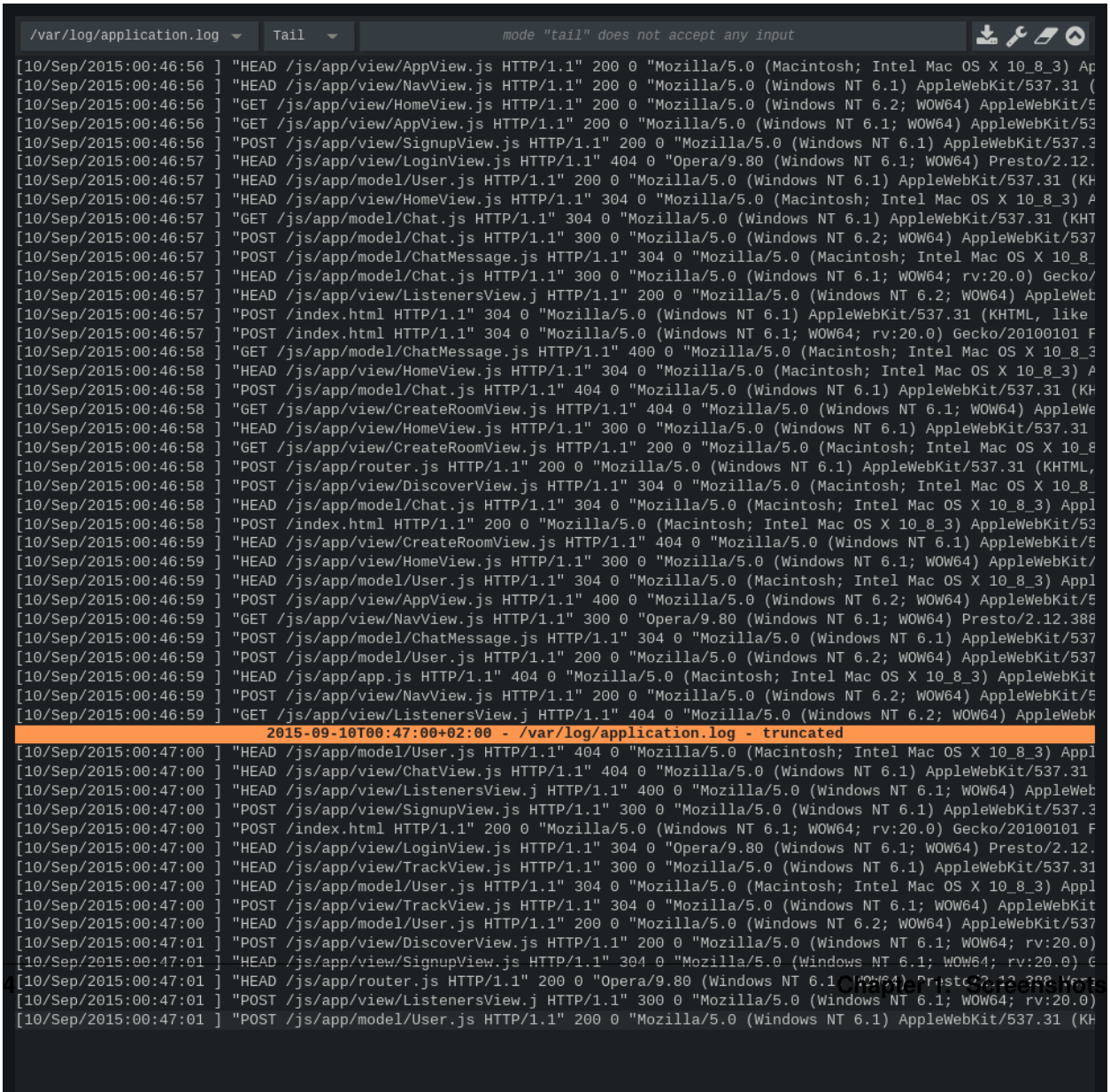
9 License **21**

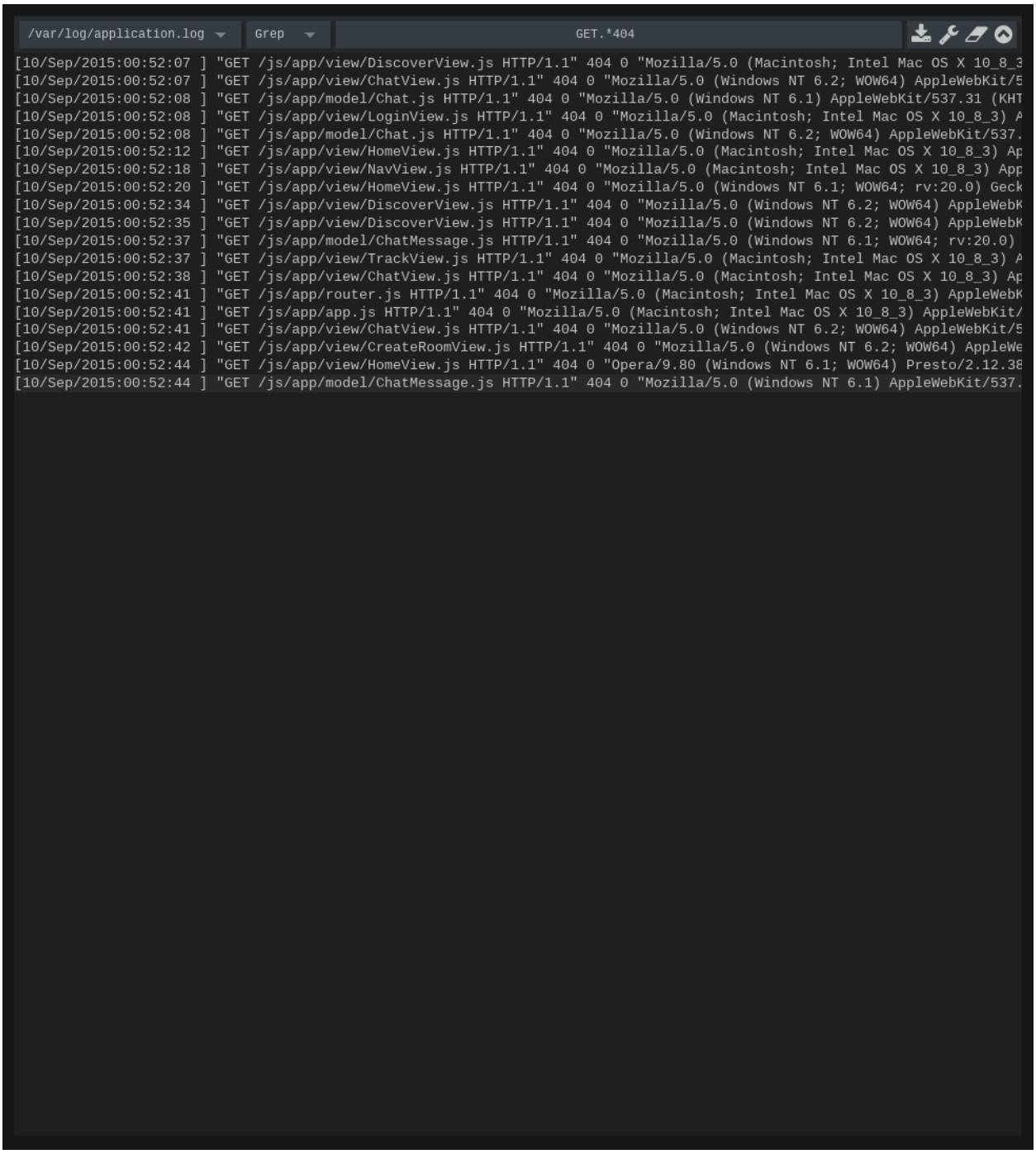
Tailon is a self-hosted web application for looking at and searching through log files. It is little more than a fancy web wrapper around the following commands:

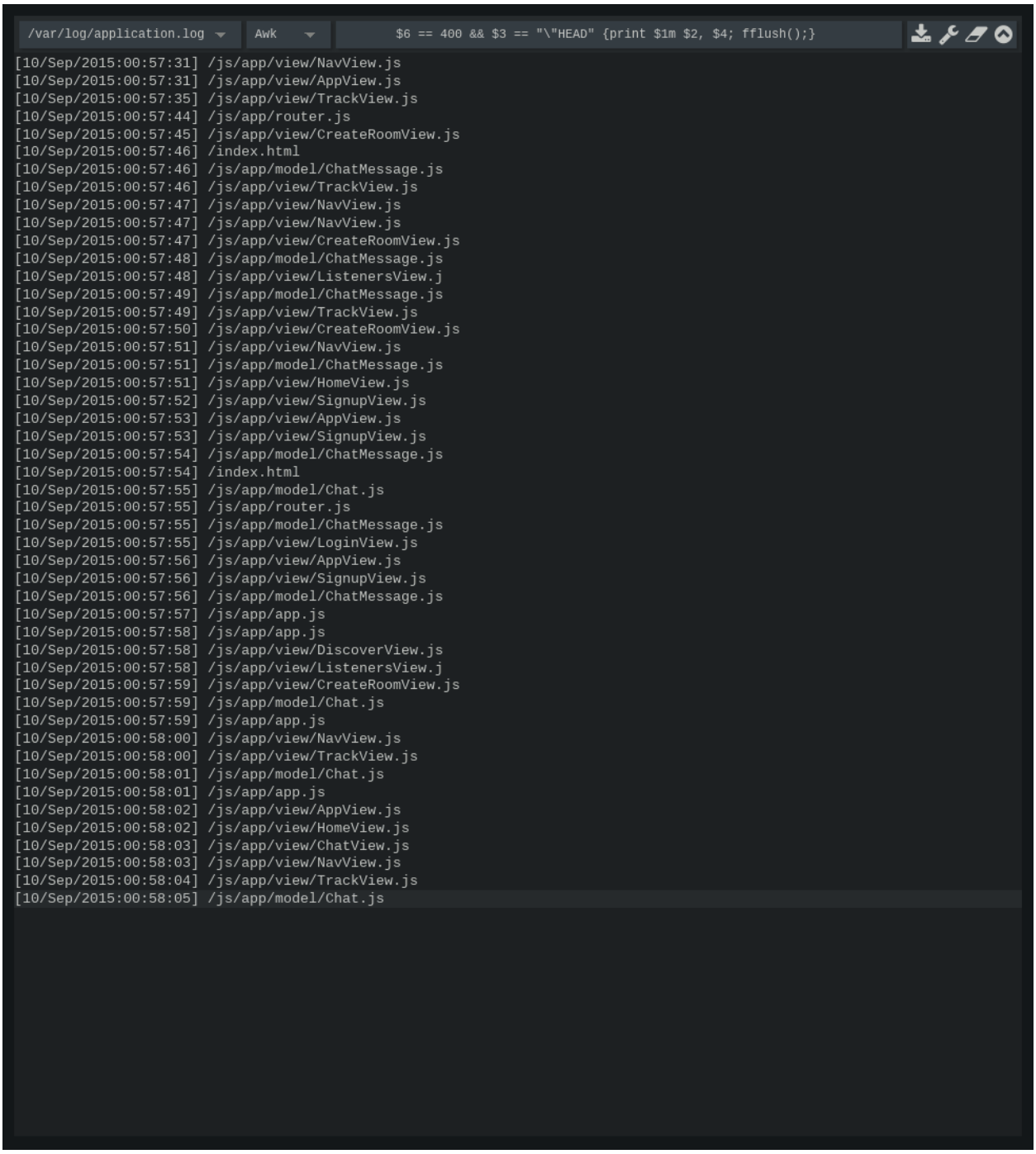
```
tail -f
tail -f | grep
tail -f | awk
tail -f | sed
```

CHAPTER 1

Screenshots




/var/log/application.log
Grep
GET.*404
[10/Sep/2015:00:52:07] "GET /js/app/view/DiscoverView.js HTTP/1.1" 404 0 "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3
[10/Sep/2015:00:52:07] "GET /js/app/view/ChatView.js HTTP/1.1" 404 0 "Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/5
[10/Sep/2015:00:52:08] "GET /js/app/model/Chat.js HTTP/1.1" 404 0 "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.31 (KHT
[10/Sep/2015:00:52:08] "GET /js/app/view/LoginView.js HTTP/1.1" 404 0 "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3) A
[10/Sep/2015:00:52:08] "GET /js/app/model/Chat.js HTTP/1.1" 404 0 "Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.
[10/Sep/2015:00:52:12] "GET /js/app/view/HomeView.js HTTP/1.1" 404 0 "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3) Ap
[10/Sep/2015:00:52:18] "GET /js/app/view/NavView.js HTTP/1.1" 404 0 "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3) App
[10/Sep/2015:00:52:20] "GET /js/app/view/HomeView.js HTTP/1.1" 404 0 "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:20.0) Geck
[10/Sep/2015:00:52:34] "GET /js/app/view/DiscoverView.js HTTP/1.1" 404 0 "Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebK
[10/Sep/2015:00:52:35] "GET /js/app/view/DiscoverView.js HTTP/1.1" 404 0 "Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebK
[10/Sep/2015:00:52:37] "GET /js/app/model/ChatMessage.js HTTP/1.1" 404 0 "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:20.0)
[10/Sep/2015:00:52:37] "GET /js/app/view/TrackView.js HTTP/1.1" 404 0 "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3) A
[10/Sep/2015:00:52:38] "GET /js/app/view/ChatView.js HTTP/1.1" 404 0 "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3) Ap
[10/Sep/2015:00:52:41] "GET /js/app/router.js HTTP/1.1" 404 0 "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3) AppleWebK
[10/Sep/2015:00:52:41] "GET /js/app/app.js HTTP/1.1" 404 0 "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3) AppleWebKit/
[10/Sep/2015:00:52:41] "GET /js/app/view/ChatView.js HTTP/1.1" 404 0 "Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/5
[10/Sep/2015:00:52:42] "GET /js/app/view/CreateRoomView.js HTTP/1.1" 404 0 "Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWe
[10/Sep/2015:00:52:44] "GET /js/app/view/HomeView.js HTTP/1.1" 404 0 "Opera/9.80 (Windows NT 6.1; WOW64) Presto/2.12.38
[10/Sep/2015:00:52:44] "GET /js/app/model/ChatMessage.js HTTP/1.1" 404 0 "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.


/var/log/application.log
Awk
$6 == 400 && $3 == "\"HEAD" {print $1m $2, $4; fflush();}
[10/Sep/2015:00:57:31] /js/app/view/NavView.js
[10/Sep/2015:00:57:31] /js/app/view/AppView.js
[10/Sep/2015:00:57:35] /js/app/view/TrackView.js
[10/Sep/2015:00:57:44] /js/app/router.js
[10/Sep/2015:00:57:45] /js/app/view/CreateRoomView.js
[10/Sep/2015:00:57:46] /index.html
[10/Sep/2015:00:57:46] /js/app/model/ChatMessage.js
[10/Sep/2015:00:57:46] /js/app/view/TrackView.js
[10/Sep/2015:00:57:47] /js/app/view/NavView.js
[10/Sep/2015:00:57:47] /js/app/view/NavView.js
[10/Sep/2015:00:57:47] /js/app/view/CreateRoomView.js
[10/Sep/2015:00:57:48] /js/app/model/ChatMessage.js
[10/Sep/2015:00:57:48] /js/app/view/ListenersView.j
[10/Sep/2015:00:57:49] /js/app/model/ChatMessage.js
[10/Sep/2015:00:57:49] /js/app/view/TrackView.js
[10/Sep/2015:00:57:50] /js/app/view/CreateRoomView.js
[10/Sep/2015:00:57:51] /js/app/view/NavView.js
[10/Sep/2015:00:57:51] /js/app/model/ChatMessage.js
[10/Sep/2015:00:57:51] /js/app/view/HomeView.js
[10/Sep/2015:00:57:52] /js/app/view/SignupView.js
[10/Sep/2015:00:57:53] /js/app/view/AppView.js
[10/Sep/2015:00:57:53] /js/app/view/SignupView.js
[10/Sep/2015:00:57:54] /js/app/model/ChatMessage.js
[10/Sep/2015:00:57:54] /index.html
[10/Sep/2015:00:57:55] /js/app/model/Chat.js
[10/Sep/2015:00:57:55] /js/app/router.js
[10/Sep/2015:00:57:55] /js/app/model/ChatMessage.js
[10/Sep/2015:00:57:55] /js/app/view/LoginView.js
[10/Sep/2015:00:57:56] /js/app/view/AppView.js
[10/Sep/2015:00:57:56] /js/app/view/SignupView.js
[10/Sep/2015:00:57:56] /js/app/model/ChatMessage.js
[10/Sep/2015:00:57:57] /js/app/app.js
[10/Sep/2015:00:57:58] /js/app/app.js
[10/Sep/2015:00:57:58] /js/app/view/DiscoverView.js
[10/Sep/2015:00:57:58] /js/app/view/ListenersView.j
[10/Sep/2015:00:57:59] /js/app/view/CreateRoomView.js
[10/Sep/2015:00:57:59] /js/app/model/Chat.js
[10/Sep/2015:00:57:59] /js/app/app.js
[10/Sep/2015:00:58:00] /js/app/view/NavView.js
[10/Sep/2015:00:58:00] /js/app/view/TrackView.js
[10/Sep/2015:00:58:01] /js/app/model/Chat.js
[10/Sep/2015:00:58:01] /js/app/app.js
[10/Sep/2015:00:58:02] /js/app/view/AppView.js
[10/Sep/2015:00:58:02] /js/app/view/HomeView.js
[10/Sep/2015:00:58:03] /js/app/view/ChatView.js
[10/Sep/2015:00:58:03] /js/app/view/NavView.js
[10/Sep/2015:00:58:04] /js/app/view/TrackView.js
[10/Sep/2015:00:58:05] /js/app/model/Chat.js

CHAPTER 2

Installation

The latest stable version of tailon can be installed from pypi:

```
$ pip install tailon
```

The development version is available on github and can also be installed with the help of pip:

```
$ pip install git+git://github.com/gvalkov/tailon.git
```

Tailon works with Python 2.7 and newer. Using it with Python >= 3.3 is encouraged.

CHAPTER 3

Quick start

3.1 Tailon

Tailon is a command-line tool that spawns a local http server that serves your logfiles. It can be configured entirely from its command-line interface or through the convenience of a yaml config file.

To get started, run tailon with the list of files that you wish to monitor:

```
$ tailon -f /var/log/nginx/* /var/log/apache/{access,error}.log
```

If at least one of the specified files is readable, tailon will start listening on http://localhost:8080.

Tailon's server-side functionality is summarized entirely in its help message:

```
Usage: tailon [-c path] [-f path [path ...]] [-h] [-d] [-v]
              [--output-encoding enc] [--input-encoding enc] [-b addr:port]
              [-r path] [-p type] [-u user:pass] [-a] [-f] [-t num]
              [-m [cmd [cmd ...]]] [--no-wrap-lines]

Tailon is a web app for looking at and searching through log files.

Required options:
  -c, --config path               yaml config file
  -f, --files path [path ...]     list of files or file wildcards to expose

General options:
  -h, --help                      show this help message and exit
  -d, --debug                     show debug messages
  -v, --version                   show program's version number and exit
  --output-encoding enc           encoding for output
  --input-encoding enc            encoding for input and output (default utf8)

Server options:
  -b, --bind addr:port            listen on the specified address and port
  -r, --relative-root path        web app root path
```

```
  -p, --http-auth type             enable http authentication (digest or basic)
  -u, --user user:pass             http authentication username and password
  -a, --allow-transfers            allow log file downloads
  -F, --follow-names               allow tailing of not-yet-existent files
  -t, --tail-lines num             number of lines to tail initially
  -m, --commands [cmd [cmd ...]]   allowed commands (default: tail grep awk)

User-interface options:
  --no-wrap-lines                  initial line-wrapping state (default: true)

Example config file:
  bind: 0.0.0.0:8080       # address and port to bind on
  allow-transfers: true    # allow log file downloads
  follow-names: false      # allow tailing of not-yet-existent files
  relative-root: /tailon   # web app root path (default: '')
  commands: [tail, grep]   # allowed commands
  tail-lines: 10           # number of lines to tail initially
  wrap-lines: true         # initial line-wrapping state

  files:
    - '/var/log/messages'
    - '/var/log/nginx/*.log'
    - '/var/log/xorg.[0-10].log'
    - '/var/log/nginx/'    # all files in this directory
    - 'cron':              # it's possible to add sub-sections
        - '/var/log/cron*'

  http-auth: basic         # enable authentication (optional)
  users:                   # password access (optional)
    user1: pass1

Example command-line:
  tailon -f /var/log/messages /var/log/debug -m tail
  tailon -f '/var/log/cron*' -a -b localhost:8080
  tailon -f /var/log/ -p basic -u user1:pass1 -u user2:pass2
  tailon -c config.yaml -d
```

Please note that if the file list includes wildcard characters, they will be expanded only once at server-start time.

CHAPTER 4

Reverse proxy configuration

4.1 Nginx

1. Run `tailon`, binding it to localhost and specifiying a relative root of your liking. For example:

```
$ tailon -f /var/log/nginx/* -b localhost:8084 -r '/tailon/'
```

2. Add the following location directives to `nginx.conf`:

```
location /tailon/ws {
    proxy_pass http://localhost:8084/tailon/ws;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
}

location /tailon {
    proxy_pass http://localhost:8084;
}
```

CHAPTER 5

Security

Tailon runs commands on the server it is installed on. While commands that accept a script argument (such as awk, sed and grep) should be invulnerable to shell injection, they may still allow for arbitrary command execution and unrestricted access to the filesystem.

To clarify this point, consider the following input to the sed command:

```
s/a/b'; cat /etc/secrets
```

This will result in an error, as tailon does not invoke commands through a shell. On the other hand, the following command is a perfectly valid sed script that has the same effect as the above attempt for shell injection:

```
r /etc/secrets
```

The default set of enabled commands - tail, grep and awk - should be safe to use. GNU awk is run in sandbox mode, which prevents scripts from accessing your system, either through the `system()` builtin or by using input redirection.

By default, tailon is accessible to anyone who knows the server address and port. One way to restrict access is by using the built-in basic and digest http authentication. This can be enabled on the command-line with:

```
$ tailon -p basic  -u joe:secret1 -u bob:secret2
$ tailon -p digest -u joe:secret1 -u bob:secret2
```

CHAPTER 6

Development

Code, bug reports and feature requests are kindly accepted on tailon's github page. Please refer to the development document for more information on developing tailon.

CHAPTER 7

Similar Projects

- clarity
- errorlog
- log.io
- rtail
- wtee

CHAPTER 8

Attributions

Tailon's favicon was created from this icon.

CHAPTER 9

License

Tailon is released under the terms of the Revised BSD License.