

目录

目录	1
Configuration	2
Configuration	2
Contributing	3
Contributing	3
Some General Rules	3
Home	4
Welcome to the joshuto wiki!	4
Tidbits	5
Exit to current directory	5
joshuto.toml	6
joshuto.toml	6
keymap.toml	7
keymap.toml	7
Keys available:	7
Commands available:	7
General	7
Navigation	8
Tabs	9
File Operations	9
Search	9
Misc	10
mimetype.toml	11
mimetype.toml	11
Class and inherit	11
Example	11
Explanation of the configuration above	11
theme.toml	13
theme.toml	13
Style	13
Color	13
Theme	13

Configuration

Configuration

Joshuto reads configurations from the following directories using environment variables (in order of precedence):

- `$JOSHUTO_CONFIG_HOME`
- `$XDG_CONFIG_HOME/joshuto`
- `$HOME/.config/joshuto`

Joshuto can currently be configured using the following files:

- [joshuto.toml](#): basic/general configurations
- [keymap.toml](#): keymapping configurations
- [mimetype.toml](#): mimetype configurations
- [theme.toml](#): theming configurations

Contributing

Contributing

Feel free to open a pull request!

Some General Rules

- Please make sure the code is formatted with `cargo fmt` to keep a consistent code style.
- Try to squash useless commits into more useful commits
 - Commits such as code formatting, code cleanups, fixing a feature that hasn't been merged can be merged into one commit for the original feature
- Try to have somewhat informative commit messages
 - Avoid messages like: fix, fmt,

Home

Welcome to the joshuto wiki!

- [Configuration](#)
- [Contributing](#)
- [Tidbits](#)

Tidbits

Exit to current directory

```
function joshuto() {  
  TIMESTAMP=$(date '+%Y%m%d-%H%M%S')  
  CWD_FILE="/tmp/$USER/joshuto-cwd-$TIMESTAMP"  
  env joshuto --last-dir "$CWD_FILE" $@  
  
  if [ -e "$CWD_FILE" ]; then  
    JOSHUTO_CWD=$(cat "$CWD_FILE")  
    rm "$CWD_FILE" && cd "$JOSHUTO_CWD"  
  fi  
}
```

joshuto.toml

joshuto.toml

This file is for general configurations.

All options available and their default values:

```
# This is for configuring how many items to reach before 'scrolling' the view
# CURRENTLY DOES NOT WORK
scroll_offset = 6

# If joshuto does not know how to open the file, it can resort to opening it via xdg settings
xdg_open = false

# Use system trash can instead of permanently removing files
use_trash = true

# The maximum file size to show a preview for
# CURRENTLY DOES NOT WORK
max_preview_size = 2097152 # 2MB

# Configurations related to the display
[display]
# Collapse the preview window when there is no preview available
collapse_preview = true
# Ratios for parent view, current view and preview
column_ratio = [1, 3, 4]
# Show borders around different views
show_borders = true
# Show hidden files
show_hidden = false
# Show file icons (requires a supporting font)
show_icons = true
# Show file previews
# CURRENTLY DOES NOT WORK
show_preview = true
# Shorten /home/$USER to ~
tilde_in_titlebar = true

# Configurations related to file sorting
[display.sort]
# Options include
# - lexical (10.txt comes before 2.txt)
# - natural (2.txt comes before 10.txt)
# - mtime
method = "natural"
# case sensitive sorting
case_sensitive = false
# show directories first
directories_first = true
# sort in reverse
reverse = false
```

keymap.toml

keymap.toml

This file is for mapping keyboard keys to commands.

```
mapcommand = [
  { keys = [ "T" ], command = "new_tab" },
  # ...
]
```

For more examples, take a look at [config/keymap.toml](#)

Keys available:

To combine keys with Ctrl and Alt, simply have `ctrl+key` / `alt+key` where `key` is a valid key.

In addition to the standard alphabet, Joshuto currently also support the following keys.

```
backspace
backtab    # this is shift+tab
arrow_left
arrow_right
arrow_up
arrow_down
home
end
page_up
page_down
delete
insert
escape
f1
f2
f3
f4
f5
f6
f7
f8
f9
f10
f11
f12
```

Commands available:

- [General](#)
- [Navigation](#)
- [Tabs](#)
- [Search](#)
- [Misc](#)

General

- `quit` : quit joshuto
 - will not quit if there are pending IO work (paste jobs)
- `force_quit` : forcefully quits joshuto
 - will *NOT* wait for IO operations to complete

- `quit_to_cwd` : quits joshuto and writes the current directory to a file specified by `--lastdir` command line option
 - works with the following bash function:

```
function joshuto() {
    TIMESTAMP=$(date '+%Y%m%d-%H%M%S')
    CWD_FILE="/tmp/$USER/joshuto-cwd-$TIMESTAMP"
    env joshuto --last-dir "$CWD_FILE" $@

    if [ -e "$CWD_FILE" ]; then
        JOSHUTO_CWD=$(cat "$CWD_FILE")
        rm "$CWD_FILE" && cd "$JOSHUTO_CWD"
    fi
}
```

- `:` : opens the command prompt
 - this does not execute the command, but merely sets the text to it
 - Example: `:cd /` will open up the command prompt with `cd /` already written
- `shell` : runs a shell command
 - Example: `:shell touch file.txt` will create a file called `file.txt`
- `spawn` : runs a shell command in the background
- `sort` : change the sort method
 - `sort lexical` : sort lexically (`10.txt` comes before `2.txt`)
 - `sort natural` : sort naturally (`2.txt` comes before `10.txt`)
 - `sort mtime` : sort via last modified time
 - `sort reverse` : reverse the sorting
- `show_workers` : show the pending IO operations and the current progress
 - press `escape` to exit view
- `toggle_hidden` : toggle hidden files

Navigation

- `cursor_move_up` : moves the cursor up by `x` amount
 - `cursor_move_up` : moves the cursor up by 1
 - `cursor_move_up x` : moves the cursor up by `x` where `x` is a non-negative number
- `cursor_move_down` : moves the cursor down by `x` amount
 - `cursor_move_down` : moves the cursor down by 1
 - `cursor_move_down x` : moves the cursor down by `x` where `x` is a non-negative number
- `cursor_move_home` : moves cursor to beginning of directory list
- `cursor_move_end` : moves cursor to end of directory list
- `cursor_move_page_up` : moves the cursor up by `x`
 - where `x` is the number of items that can be seen on the screen
- `cursor_move_page_down` : moves the cursor down by `x`
 - where `x` is the number of items that can be seen on the screen
- `parent_cursor_move_up`
 - same as `cursor_move_up` but moves the cursor in the parent list instead of the current list
- `parent_cursor_move_down`

- same as `cursor_move_down` but moves the cursor in the parent list instead of the current list
- `cd` : change directory
 - `cd ..` : go to parent directory
 - `cd ~` : go to home directory
- `open` : open file or directory
 - if joshuto does not know how to open the file format (via extension currently), it will prompt `:open_with` to open with a specific command
 - if `xdg_open` is `true` in [joshuto.toml](#), joshuto will try to open it via xdg settings
- `subdir_fzf` : go to a subdirectory via fzf

Tabs

- `new_tab` : opens a new tab
 - new tab opens in home directory
- `close_tab` : close current tab
- `tab_switch` : switch to next/previous tab by `x`
 - where `x` is an integer
 - `tab_switch 1` : go to next tab
 - `tab_switch -1` : go to previous tab

File Operations

- `reload_dirlist` : reloads the current directory listing
- `mkdir` : create a new directory (usually used as `:mkdir`)
- `cut_files` : store selected files (or current file if none were selected) to be moved later
- `copy_files` : store selected files (or current file if none were selected) to be copied later
- `paste_files` : move/copy files stored from a previous `cut_files` or `copy_files` command
- `delete_files` : delete selected files (or current file if none were selected).
 - will *permanently* delete files if `use_trash` is `false` in [joshuto.toml/wiki/Configuration#joshutotoml](#)
 - if `use_trash` is `true` , this might cause issues with deleting files on mounted filesystems such as on an external hard drive or tmpfs
- `rename` : rename current file
 - `:rename new_name`
- `rename_append` : opens the command prompt with the rename command and the current file name filled in.
 - cursor will be set right before the extension of the file (end of file name if no extension)
- `rename_prepend` : opens the command prompt with the rename command and the current file name filled in.
 - cursor will be set to the beginning of the file name
- `copy_filename` : copy the file name to clipboard
 - clipboard support requires xsel, xclip, or wl-copy
- `copy_filename_without_extension` : copy the file name without the extension to clipboard
- `copy_filepath` : copy the file path to clipboard
- `copy_dirpath` : copy the current directory path to clipboard
- `set_mode` : Set read, write, execute permissions of current file
- `touch` : create a new file or update an existing file's modified time

Search

- `search` : search via string
 - case insensitive
- `search_glob` : search via shell globbing
 - `:search_glob *.png`
- `search_fzf` : search via fzf
- `search_next` : go to next search result
- `search_prev` : go to previous search result
- `select` : select current file
 - `--toggle=true` : toggle the selected files
 - `--all=true` : selected all files
 - `--deselect=true` : deselect rather than select
 - `glob` : select files based on glob (just like `search_glob`)
 - `select *.png`

Misc

- `bulk_rename` : rename all selected files
 - this will create a file inside `/tmp` and open up your text editor of choice via `$EDITOR` environment variable
 - once you've made your changes to the file, saved and quit, it will use the `mv` command to rename everything

mimetype.toml

mimetype.toml

This file tells joshuto what programs to use when opening files.

There are currently 2 ways to configure opening files:

- via extension
- via mimetype (*CURRENTLY DOES NOT WORK*)

Class and inherit

To alleviate the lack of variables and programmability in TOML, there is a section for users to specify "classes" called `[class]`. Here, users can specify a list of commands to open a file and inherit these for a specific file format.

Example

```
[class]
image_default = [
  { command = "qimgv", fork = true, silent = true },
  { command = "krita", fork = true, silent = true } ]

[extension]
# inherit from image_default class
png.inherit = "image_default"

# inherit from image_default class
jpg.inherit = "image_default"
# in addition, also add gimp for .jpg files only
jpg.app_list = [
  { command = "gimp", fork = true, silent = true } ]

mkv.app_list = [
  { command = "mpv", args = [ "--" ], fork = true, silent = true },
  { command = "mediainfo", confirm_exit = true },
  { command = "mpv", args = [ "--mute", "on", "--" ], fork = true, silent = true } ]
rs.app_list = [
  { command = "micro" },
  { command = "gedit", fork = true, silent = true },
  { command = "bat", confirm_exit = true } ]
```

each extension has the following fields:

- `inherit` : string indicating the class to inherit from, if any
- `app_list` : list of commands

each command has the following fields:

- `command` : the command to run
- `args` : (optional) list of arguments for the command
- `fork` : tells joshuto to run the program in foreground or background
 - foreground will pause joshuto
- `silent` : tells joshuto to discard all output of the program
 - useful when the program outputs debug messages into the terminal, potentially ruining joshuto's UI
- `confirm_exit` : requires the user's input before going back to joshuto
 - useful when you want to read the output of the command

Explanation of the configuration above

For files with `.png` extension, joshuto opens them with `qimgv`. Joshuto suppresses all terminal output from `qimgv` to prevent UI disturbance. Joshuto forks `qimgv` so we can continue using joshuto while viewing the image. Alternatively, we can open it with `krita` or `mediainfo` via `:open_with 1` and `:open_with 2`. With `mediainfo`, we want to see the output of the command before going back to joshuto, so we have

mimetype.toml

```
confirm_exit = true
```

For files with `.rs` extension, joshuto will open it with `micro`, a command line text editor. In order for joshuto and `micro` to not conflict, joshuto will wait for `micro` to exit, before redrawing. Joshuto will also not suppress `micro`'s output.

theme.toml

theme.toml

This file is used to theme joshuto

Style

Each style has the following fields:

```
# background color
bg = "light_blue"
# foreground color
fg = "blue"
bold = false
underline = false
invert = false
```

Color

Joshuto supports 16 colors as well as hex colors via `rgb(r,g,b)`

```
black
red
blue
green
yellow
magenta
cyan
white
gray
dark_gray
light_red
light_green
light_yellow
light_blue
light_magenta
light_cyan
-
```

Theme

joshuto supports theming via system file types as well as extensions

System file types include:

```
# for selected files
[selection]
fg = "light_yellow"
bold = true

# for executable files
[executable]
fg = "light_green"
bold = true

# default theme
[regular]
fg = "white"

# for directories
[directory]
fg = "light_blue"
bold = true
```

```
# for symlinks
[link]
fg = "cyan"
bold = true

# for invalid symlinks
[link_invalid]
fg = "red"
bold = true

# for sockets
[socket]
fg = "light_magenta"
bold = true
```

Via extensions

```
[ext]
[ext.jpg]
fg = "yellow"
[ext.jpeg]
fg = "yellow"
[ext.png]
fg = "yellow"
```

