buku documentation

Arun Prakash Jana

May 03, 2020

User guide

Bookmark manager like a text-based mini-web.

CHAPTER 1

Introduction

`buku` is a powerful bookmark manager written in Python3 and SQLite3. When I started writing it, I couldn't find a flexible command-line solution with a private, portable, merge-able database along with seamless GUI integration. Hence, `buku` (after my son's nickname, meaning *close to the heart* in my language).

bukuserver exposes a browsable front-end on a local web host server.

`buku` can auto-import bookmarks from your browser(s) or fetch the title and description of a bookmarked url from the web. You can use your favourite editor to compose and update bookmarks. With multiple search options, including regex and a deep scan mode (particularly for URLs), it can find any bookmark instantly. `buku` can look up the latest snapshot of a broken link on the Wayback Machine. There's an Easter egg to revisit random forgotten bookmarks too! *Buku* is too busy to track you: no hidden history, obsolete records, usage analytics or homing.

To get started right away, jump to the *Quickstart* section. We have one of the best documentation around. You'll find handy examples in the man page too. For more details, please refer to the wiki page on operational notes.

There are several *projects based on* `buku`, including a browser plug-in.

Love smart and efficient utilities? Explore my repositories. Buy me a cup of coffee if they help you.

CHAPTER 2

Table of Contents

CHAPTER 3

Features

- Store bookmarks with auto-fetched title, tags and description
- Auto-import from Firefox, Google Chrome and Chromium
- Open bookmarks and search results in browser
- Shorten, expand URLs, browse cached page from Wayback Machine
- Text editor integration
- Lightweight, clean interface, custom colors
- Powerful search options (regex, substring. . .)
- Continuous search with on the fly mode switch
- Portable, merge-able database to sync between systems
- Import/export bookmarks from/to HTML, Markdown or Orgfile
- Smart tag management using redirection (>>, >, <<)
- Multithreaded full DB refresh, manual encryption support
- Shell completion scripts, man page with handy examples
- Privacy-aware (no unconfirmed user data collection)

CHAPTER 4

Installation

4.1 Dependencies

| Feature | Dependency | | — | — | | Scripting language | Python 3.5+ | | HTTPS | certifi, urllib3 | | Encryption | cryptography | | HTML | beautifulsoup4, html5lib |

To install package dependencies using pip3, run:

```
# pip3 install certifi urllib3 cryptography beautifulsoup4
```

or on Ubuntu:

```
# apt-get install ca-certificates python3-urllib3 python3-cryptography python3-bs4
```

To copy url to clipboard at the prompt `buku` uses `xsel` (or `xclip`) on Linux, `pbcopy` (default installed) on OS X, `clip` (default installed) on Windows, `termux-clipboard` on Termux (terminal emulation for Android). If X11 is missing, GNU Screen or tmux copy-paste buffers are recognized.

4.2 From a package manager

Install `buku` from your package manager. If the version available is dated try an alternative installation method.

4.3 Release packages

Auto-generated packages (with only the cli component) for Arch Linux, CentOS, Debian, Fedora, openSUSE Leap and Ubuntu are available with the latest stable release.

NOTE: CentOS may not have the python3-beautifulsoup4 package in the repos. Install it using pip3.

4.4 From source

If you have git installed, clone this repository. Otherwise download the latest stable release or development version (*risky*).

Install the cli component to default location (`/usr/local`):

```
$ sudo make install
```

To remove, run:

```
$ sudo make uninstall
```

`PREFIX` is supported, in case you want to install to a different location.

4.5 Running standalone

`buku` is a standalone utility. From the containing directory, run:

```
$ chmod +x buku
$ ./buku
```

CHAPTER 5

Shell completion

Shell completion scripts for Bash, Fish and Zsh can be found in respective subdirectories of auto-completion/. Please refer to your shell's manual for installation instructions.

CHAPTER 6

Usage

6.1 Command-line options

```
usage: buku [OPTIONS] [KEYWORD [KEYWORD ...]]

Bookmark manager like a text-based mini-web.

POSITIONAL ARGUMENTS:
      KEYWORD               search keywords

GENERAL OPTIONS:
      -a, --add URL [tag, ...]
                            bookmark URL with comma-separated tags
      -u, --update [...]    update fields of an existing bookmark
                            accepts indices and ranges
                            refresh title and desc if no edit options
                            if no arguments:
                            - update results when used with search
                            - otherwise refresh all titles and desc
      -w, --write [editor|index]
                            open editor to edit a fresh bookmark
                            edit last bookmark, if index=-1
                            to specify index, EDITOR must be set
      -d, --delete [...]    remove bookmarks from DB
                            accepts indices or a single range
                            if no arguments:
                            - delete results when used with search
                            - otherwise delete all bookmarks
      -h, --help            show this information and exit
      -v, --version         show the program version and exit

EDIT OPTIONS:
      --url keyword         bookmark link
      --tag [+|-] [...]     comma-separated tags
```

(continues on next page)

(continued from previous page)

```
                            clear bookmark tagset, if no arguments
                            '+' appends to, '-' removes from tagset
      --title [...]         bookmark title; if no arguments:
                            -a: do not set title, -u: clear title
      -c, --comment [...]   notes or description of the bookmark
                            clears description, if no arguments
      --immutable N         disable web-fetch during auto-refresh
                            N=0: mutable (default), N=1: immutable

SEARCH OPTIONS:
      -s, --sany [...]      find records with ANY matching keyword
                            this is the default search option
      -S, --sall [...]      find records matching ALL the keywords
                            special keywords -
                            "blank": entries with empty title/tag
                            "immutable": entries with locked title
      --deep                match substrings ('pen' matches 'opens')
      -r, --sreg expr       run a regex search
      -t, --stag [tag [,|+] ...] [- tag, ...]
                            search bookmarks by tags
                            use ',' to find entries matching ANY tag
                            use '+' to find entries matching ALL tags
                            excludes entries with tags after ' - '
                            list all tags, if no search keywords
      -x, --exclude [...]   omit records matching specified keywords

ENCRYPTION OPTIONS:
      -l, --lock [N]        encrypt DB in N (default 8) # iterations
      -k, --unlock [N]      decrypt DB in N (default 8) # iterations

POWER TOYS:
      --ai                  auto-import from Firefox/Chrome/Chromium
      -e, --export file     export bookmarks to Firefox format HTML
                            export Markdown, if file ends with '.md'
                            format: [title](url) <!-- TAGS -->
                            export Orgfile, if file ends with '.org'
                            format: *[[url][title]] :tags:
                            export buku DB, if file ends with '.db'
                            combines with search results, if opted
      -i, --import file     import bookmarks based on file extension
                            supports 'html', 'json', 'md', 'org', 'db'
      -p, --print [...]     show record details by indices, ranges
                            print all bookmarks, if no arguments
                            -n shows the last n results (like tail)
      -f, --format N        limit fields in -p or JSON search output
                            N=1: URL; N=2: URL, tag; N=3: title;
                            N=4: URL, title, tag; N=5: title, tag;
                            N0 (10, 20, 30, 40, 50) omits DB index
      -j, --json            JSON formatted output for -p and search
      --colors COLORS       set output colors in five-letter string
      --nc                  disable color output
      -n, --count N         show N results per page (default 10)
      --np                  do not show the prompt, run and exit
      -o, --open [...]      browse bookmarks by indices and ranges
                            open a random bookmark, if no arguments
      --oa                  browse all search results immediately
      --replace old new     replace old tag with new tag everywhere
```

(continues on next page)

(continued from previous page)

```
                            delete old tag, if new tag not specified
      --shorten index|URL   fetch shortened url from tny.im service
      --expand index|URL    expand a tny.im shortened url
      --cached index|URL    browse a cached page from Wayback Machine
      --suggest             show similar tags when adding bookmarks
      --tacit               reduce verbosity
      --threads N           max network connections in full refresh
                            default N=4, min N=1, max N=10
      -V                    check latest upstream version available
      -z, --debug           show debug information and verbose logs

SYMBOLS:
      >                     url
      +                     comment
      #                     tags

PROMPT KEYS:
    1-N                     browse search result indices and/or ranges
    O [id|range [...]]      open search results/indices in GUI browser
                            toggle try GUI browser if no arguments
    a                       open all results in browser
    s keyword [...]         search for records with ANY keyword
    S keyword [...]         search for records with ALL keywords
    d                       match substrings ('pen' matches 'opened')
    r expression            run a regex search
    t [tag, ...]            search by tags; show taglist, if no args
    g taglist id|range [...] [>>|>|<<] [record id|range ...]
                            append, set, remove (all or specific) tags
                            search by taglist id(s) if records are omitted
    n                       show next page of search results
    o id|range [...]        browse bookmarks by indices and/or ranges
    p id|range [...]        print bookmarks by indices and/or ranges
    w [editor|id]           edit and add or update a bookmark
    c id                    copy url at search result index to clipboard
    ?                       show this help
    q, ^D, double Enter     exit buku
```

6.2 Colors

`buku` supports custom colors. Visit the wiki page on how to customize colors for more details.

CHAPTER 7

Quickstart

1. Export `VISUAL` or `EDITOR` to point to your favourite editor. Note that `VISUAL` takes precedence over `EDITOR`.
2. Create a sweeter shortcut with some convenience.

```
alias b='buku --suggest'
```

3. Auto-import bookmarks from your browser(s). Please quit the relevant browsers beforehand to ensure the databases are not locked.

```
b --ai
```

4. Manually add a bookmark (for hands-on).

```
b -w
```

5. List your bookmarks with DB index.

```
b -p
```

6. For GUI and browser integration (or to sync bookmarks with your favourite bookmark management service) refer to the wiki page on System integration.

CHAPTER 8

Examples

1. **Edit and add** a bookmark from editor:

```
$ buku -w
$ buku -w 'gedit -w'
$ buku -w 'macvim -f' -a https://ddg.gg search engine, privacy
```

   The first command picks editor from the environment variable `EDITOR`. The second command opens gedit in blocking mode. The third command opens macvim with option -f and the URL and tags populated in template.

2. **Add** a bookmark with **tags** `search engine` and `privacy`, **comment** `Search engine with perks`, **fetch page title** from the web:

```
$ buku -a https://ddg.gg search engine, privacy -c Search engine with perks
336. DuckDuckGo
> https://ddg.gg
+ Alternative search engine with perks
# privacy,search engine
```

   where, >: url, +: comment, #: tags

3. **Add** a bookmark with tags `search engine` & `privacy` and **immutable custom title** `DDG`:

```
$ buku -a https://ddg.gg search engine, privacy --title 'DDG' --immutable 1
336. DDG (L)
> https://ddg.gg
# privacy,search engine
```

   Note that URL must precede tags.

4. **Add** a bookmark **without a title** (works for update too):

```
$ buku -a https://ddg.gg search engine, privacy --title
```

5. **Edit and update** a bookmark from editor:

```
$ buku -w 15012014
```

This will open the existing bookmark's details in the editor for modifications. Environment variable `EDITOR` must be set.

6. **Update** existing bookmark at index 15012014 with new URL, tags and comments, fetch title from the web:

```
$ buku -u 15012014 --url http://ddg.gg/ --tag web search, utilities -c Private␣
↪search engine
```

7. **Fetch and update only title** for bookmark at 15012014:

```
$ buku -u 15012014
```

8. **Update only comment** for bookmark at 15012014:

```
$ buku -u 15012014 -c this is a new comment
```

Applies to –url, –title and –tag too.

9. **Export** bookmarks tagged `tag 1` or `tag 2` to HTML, Markdown, Orgfile or a new database:

```
$ buku -e bookmarks.html --stag tag 1, tag 2
$ buku -e bookmarks.md --stag tag 1, tag 2
$ buku -e bookmarks.org --stag tag 1, tag 2
$ buku -e bookmarks.db --stag tag 1, tag 2
```

All bookmarks are exported if search is not opted.

10. **Import** bookmarks from HTML, Markdown or Orgfile:

```
$ buku -i bookmarks.html
$ buku -i bookmarks.md
$ buku -i bookmarks.org
$ buku -i bookmarks.db
```

11. **Delete only comment** for bookmark at 15012014:

```
$ buku -u 15012014 -c
```

Applies to –title and –tag too. URL cannot be deleted without deleting the bookmark.

12. **Update** or refresh **full DB** with page titles from the web:

```
$ buku -u
$ buku -u --tacit (show only failures and exceptions)
```

This operation can update the title or description fields of non-immutable bookmarks by parsing the fetched page. Fields are updated only if the fetched fields are non-empty. Tags remain untouched.

13. **Delete** bookmark at index 15012014:

```
$ buku -d 15012014
Index 15012020 moved to 15012014
```

The last index is moved to the deleted index to keep the DB compact.

14. **Delete all** bookmarks:

```
$ buku -d
```

15. **Delete** a **range or list** of bookmarks:

```
$ buku -d 100-200
$ buku -d 100 15 200
```

16. **Search** bookmarks for **ANY** of the keywords `kernel` and `debugging` in URL, title or tags:

```
$ buku kernel debugging
$ buku -s kernel debugging
```

17. **Search** bookmarks with **ALL** the keywords `kernel` and `debugging` in URL, title or tags:

```
$ buku -S kernel debugging
```

18. **Search** bookmarks **tagged** `general kernel concepts`:

```
$ buku --stag general kernel concepts
```

19. **Search** for bookmarks matching **ANY** of the tags `kernel`, `debugging`, `general kernel concepts`:

```
$ buku --stag kernel, debugging, general kernel concepts
```

20. **Search** for bookmarks matching **ALL** of the tags `kernel`, `debugging`, `general kernel concepts`:

```
$ buku --stag kernel + debugging + general kernel concepts
```

21. **Search** for bookmarks matching any of the keywords `hello` or `world`, excluding the keywords `real` and `life`, matching both the tags `kernel` and `debugging`, but **excluding** the tags `general kernel concepts` and `books`:

```
$ buku hello world --exclude real life --stag 'kernel + debugging - general␣
↪kernel concepts, books'
```

22. List **all unique tags** alphabetically:

```
$ buku --stag
```

23. Run a **search and update** the results:

```
$ buku -s kernel debugging -u --tag + linux kernel
```

24. Run a **search and delete** the results:

```
$ buku -s kernel debugging -d
```

25. **Encrypt or decrypt** DB with **custom number of iterations** (15) to generate key:

```
$ buku -l 15
$ buku -k 15
```

    The same number of iterations must be specified for one lock & unlock instance. Default is 8, if omitted.

26. **Show details** of bookmarks at index 15012014 and ranges 20-30, 40-50:

```
$ buku -p 20-30 15012014 40-50
```

27. Show details of the **last 10 bookmarks**:

```
$ buku -p -10
```

28. **Show all** bookmarks with real index from database:

```
$ buku -p
$ buku -p | more
```

29. **Replace tag** 'old tag' with 'new tag':

```
$ buku --replace 'old tag' 'new tag'
```

30. **Delete tag** 'old tag' from DB:

```
$ buku --replace 'old tag'
```

31. **Append (or delete) tags** 'tag 1', 'tag 2' to (or from) existing tags of bookmark at index 15012014:

```
$ buku -u 15012014 --tag + tag 1, tag 2
$ buku -u 15012014 --tag - tag 1, tag 2
```

32. **Open URL** at index 15012014 in browser:

```
$ buku -o 15012014
```

33. List bookmarks with **no title or tags** for bookkeeping:

```
$ buku -S blank
```

34. List bookmarks with **immutable title**:

```
$ buku -S immutable
```

35. **Shorten URL** www.google.com and the URL at index 20:

```
$ buku --shorten www.google.com
$ buku --shorten 20
```

36. **Append, remove tags at prompt** (taglist index to the left, bookmark index to the right):

```
// append tags at taglist indices 4 and 6-9 to existing tags in bookmarks at␣
↪indices 5 and 2-3
buku (? for help) g 4 9-6 >> 5 3-2
// set tags at taglist indices 4 and 6-9 as tags in bookmarks at indices 5 and 2-3
buku (? for help) g 4 9-6 > 5 3-2
// remove all tags from bookmarks at indices 5 and 2-3
buku (? for help) g > 5 3-2
// remove tags at taglist indices 4 and 6-9 from tags in bookmarks at indices 5␣
↪and 2-3
buku (? for help) g 4 9-6 << 5 3-2
```

37. List bookmarks with **colored output**:

```
$ buku --colors oKlxm -p
```

38. More **help**:

```
$ buku -h
$ man buku
```

CHAPTER 9

Automation

Interactive workflows can be automated using expect. Issue #368 has a working example on automating auto-import.

CHAPTER 10

Troubleshooting

10.1 Editor integration

You may encounter issues with GUI editors which maintain only one instance by default and return immediately from other instances. Use the appropriate editor option to block the caller when a new document is opened. See issue #210 for gedit.

CHAPTER 11

Collaborators

- Arun Prakash Jana
- Rachmadani Haryono
- Johnathan Jenkins
- SZ Lin

CHAPTER 12

Contributions

Missing a feature? There's a rolling ToDo List with identified tasks. Contributions are welcome! Please follow the PR guidelines.

CHAPTER 13

Related projects

- bukubrow, WebExtension for browser integration
- oil, search-as-you-type cli front-end
- buku_run, rofi front-end
- pinku, a Pinboard-to-buku import utility
- buku-dmenu, a simple bash dmenu wrapper
- poku, sync between Pocket and buku
- Ebuku, Emacs interface to buku

CHAPTER 14

In the Press

- 2daygeek
- Hacker Milk
- It's F.O.S.S.
- LinOxide
- LinuxUser Magazine 01/2017 Issue
- Make Tech Easier
- One Thing Well
- Open Source For You
- ulno.net

CHAPTER 15

buku module

CHAPTER 16

Indices and tables

- genindex
- modindex
- search