Wallypub User Guide

Wallypub is a CLI companion tool for Wallabag that allows you to turn Wallabag articles into a dedicated epub.
This guide provides in-depth information about how to work with this tool.

- Introduction
- Usage
  - Commands
- Setup
  - Installation
  - Settings
  - Secrets
  - Troubleshooting
- Issues
- Frequently Asked Questions

Introduction

Supported systems and software intent.

Supported Systems

Linux is the primary target for this tool. The burden of ensuring this works on Windows & MacOS is too great. This CLI tool was built against Arch and Ubuntu distributions. If there are issues on other Linux distributions, please file an issue

Design Goals

- Slow down news consumption by building a way to create contained digests that encourages people to actually read the articles in their read it later service.
- Ensure that if you use this CLI tool, a person does not need to muck about in their file system if they do not want to.
- Purposefully use flat files for configuration so that tools outside of Wallypub can be used.

Usage

Commands

The commands are present in alphabetical order and organized by the domain followed by the action. Most of the information can be found through the use of the `--help` flag. This page provides information on nuances and edge cases that might not have fit in the CLI descriptions themselves.

```
Usage: main.py [OPTIONS] COMMAND [ARGS]...

Options:
  --help  Show this message and exit.

Commands:
  config-edit          prompts user to update settings values
  config-init          walks the user through the first time configuration
  config-secrets-edit  prompts user to update secret values in keyring
  config-secrets-show  displays the sensitive variables from the keyring
  config-show          displays the settings to the console
  digest-by-ids        generate EPUB from comma-separated Wallabag IDs
  digest-cover-create  procedurally generate a cover image for your digest
  digest-recent        generate EPUB from recent Wallabag articles
  donate               prints link to Glass Hound Computing's donation page
  entry-add            add an entry to your Wallabag instance
  system-fonts-show    displays installed system fonts
```

config-edit

prompts user to update settings values

An in depth guide about the configuration can be found in the settings section.

config-init

walks the user through the first time configuration

The first time configuration assumes some defaults, opting to only setup the requirements to generate a digest. Additional settings can be adjusted through the config-edit command.

If the default font is not installed, wallypub will install that in your system fonts.

config-secrets-edit

prompts user to update secret values in keyring

More details about the secrets can be found in the secrets section.

config-secrets-show

displays the sensitive variables from the keyring

config-show

displays the settings to the console

digest-by-ids

generate EPUB from comma-separated Wallabag IDs

The IDs should correlate with Wallabag entries. When viewing from the web, the entry ID can be found in the URL In https://app.wallabag.it/view/32247628 the ID would be 32247628

digest-cover-create

procedurally generate a cover image for your digest

digest_cover_create exists mostly to have a means of independently test that image generation works on the machine. It is also useful for testing new fonts to see how they look on the page.

Wallypub will automatically generate a new cover if the configured filename in the cover_file setting matches the default name of "cover.jpg", this remains true even if you configure another directory as the location for the static files. Recommendation is to call the file something "custom_cover.ext" or "name_of_digest.ext".

digest-recent

generate EPUB from recent Wallabag articles

donate

prints link to Glass Hound Computing's donation page

buymeacoffee.com/glasshoundcomputing

entry-add

```
Usage: main.py entry-add [OPTIONS] URL

  add an entry to your Wallabag instance

  The `--tags` flag for `entry-add` needs to be wrapped in a string `""` to
  get comma separation if you intend on using spaces
  within your tags.

Options:
  --tags TEXT  comma separated string
  --help       Show this message and exit.
```

system-fonts-show

```
Usage: main.py system-fonts-show [OPTIONS]

  displays installed system fonts

Options:
  --scroll       allows you to scroll through system fonts q to quit
  --search TEXT  fuzzy search for font names
  --help         Show this message and exit.
```

Setup

Installation, settings, and secrets

Installation

Supported installation methods.

If installing through the AUR or the binary, the first time run will take longer since these rely on PyApp to run.

Note: after installation, it is encouraged to run `config-init` for the first-time setup.

Arch

```
yay -S wallypub
```

Binary

1. Download

```
curl -LO https://codeberg.org/fausty/wallypub/releases/download/1.0.1/wallypub_1.0.1
```

2. Move to usr bin

```
sudo mv wallypub_1.0.1 /usr/local/bin/wallypub
```

3. Make executable

```
sudo chmod +x /usr/local/bin/wallypub
```

4. Run

```
wallypub --help
```

Python

https://pypi.org/project/wallypub/

```
pip install wallypub
```

Settings

Initial Configuration

Running `config-init` will take you through your first-time setup. During the initial configuration, Wallypub only asks for the bare minimum information to create a digest. This includes the name of the digest, who you would like to attribute its authorship to, and your Wallabag API credentials.

If you do not have the default system font installed, Wallypub will prompt you for your sudo password to copy the default font, `FreeMono.ttf` into the system fonts location `/usr/share/fonts`.

There are more settings, explained below, and these can be edited after the first time setup by running `config-edit`.

Digest Settings

`digest_title`: This is the title of the epub that will appear in the file name and on the cover, if you use an image generated by Wallypub. The default title is `Wallypub Presents`.

`digest_author`: This will show up on the title page and in the metadata. You can leave this blank without issue; it defaults to "".

`back_matter`: This generates the final page in your EPUB. The back matter page just credits Wallypub and provides a link to donate. This can be disabled by setting this value to False.

File locations for creating a digest

`digest_directory_filepath`: Location of files for creating the digest, defaults to `~/.local/share/Wallypub/static"`
`digest_cover_image`: The image file that Wallypub looks for defaults to `~/.local/share/Wallypub/static/cover.jpg"`
`digest_output_path`

: Where Wallypub outputs a generated epub, defaults to `~/Documents`

Wallabag settings

`wallabag_url`: The URL where your Wallabag is hosted, defaults to `app.wallabag.it`
`wallabag_userame`: The username for your Wallabag instance.
`wallabag_password`: Your Wallabag instance password. Stored in your machine's keyring.

`wallabag_client_id`: client_id for accessing API. Follow the client setup if you do not know what this is.

`wallabag_client_secret`: client_secret for accessing API. Follow the client setup if you do not know what this is. Stored in your machine's keyring.

Retrieval Settings

The Wallabag allows one to customize which files are retrieved by passing certain URL parameters.

Article Parameters

The default configuration for article parameters assumes that the bulk of the material desired for reading is the more recently added items to Wallabag.

Example:

```
archive = "0"
starred = "0"
sort = "created"
order = "desc"
page = "1"
perPage = "10"
tags = ""
since = ""
public = ""
detail = "full"
domain_name = ""
```

The canonical place to determine the behavior of the URL parameters is in the Wallabag docs

Additional Article Parameters

Wallypub automatically filters out 0 length articles. This means that the number of recent articles you intended to make a digest from is lower. To ensure that the proper number of articles are inlcuded in the epub, a check is made and then articles are fetched using different parameters. The

default is configured to pull this article from the back of the queue. This choice can be amended in the settings, however; the default was decided as a method to add variety. It is likely that articles saved from week to week might relate to a particular big news event or a rabbit hole. This default configuration attempts to provide some variety to the digest.

Other Settings

`max_read_time` articles with a longer read time than this will be skipped. Defaults to 120 (minutes). 120 minutes is about 15000 words, depending on one's reading spead.

`minimum_reading_time` Only article with a read time greater than this number are included in the digest. Defaults to 0. 0 time articles are always filtered out because it normally indicates that Wallabag was not able to properly pull the text.

`log_level`

log_level is an `int` value that gets translated to something human-readable in the logging libraries. The values are as follows: .

```
CRITICAL = 50
ERROR = 40
WARNING = 30
INFO = 20
DEBUG = 10
NOTSET = 0
```

Note that FATAL is not present. That is because exposing that value through configuration was more difficult and unnecessary. If you have never set log levels before, DEBUG will be noisier than INFO, INFO will be noisier than WARNING, and so on. If you want a quiet terminal, increase the log level; if you are troubleshooting, decrease it.

Fonts

The fonts available are the ones on your system. If you want the cover image to be different, change the font in the settings. You can open-source fonts at The League of Moveable Type and Open Source Foundry.

To see what fonts are available on your system, you can run the show-sysetm-fonts command from wallypub.

If you prefer to use system tools for font exploration, you can also use the fc-list command from your terminal.

Fonts are stored under system fonts at `/usr/share/fonts` this location was chosen because it appears to be consistent across many distributions of Linux.

Example Settings File

On your system your settings file can be found at `~/.config/Wallypub/settings.toml` or you can use the Wallypub `config-show` to see the current configuration.

```
[AdditionalArticleParameters]
archive = "0"
starred = "0"
sort = "asc"
order = ""
page = ""
perPage = ""
tags = ""
since = ""
public = ""
detail = ""
domain_name = ""

[ArticleParameters]
archive = "0"
starred = "0"
sort = "created"
order = "desc"
page = "1"
perPage = "10"
tags = ""
since = ""
public = ""
detail = "full"
domain_name = ""

[Base]
application_directory = "/home/$USER/.local/share/Wallypub"
```

```
log_level = 20
max_read_time = 120
minimum_read_time = 0

[Digest]
title = "Wallypub Presents"
filepath = "/home/$USER/.local/share/Wallypub/static"
author = "$AUTHOR"
output_path = "/home/$USER/Documents/Wallypub"
cover_file = "cover.jpg"
title_font = "FreeMono.ttf"
date_font = "FreeMono.ttf"
back_matter = True

[Wallabag]
client_id = "$CLIENT_ID"
username = "$USER"
url = "app.wallabag.it"
```

Secrets

In order to access your Wallabag instance, Wallypub relies on two secrets. Your `wallabag_password` and your `wallabag_client_secret`.

Viewing & Editing

You are prompted to add your secrets when running the initial `config-init`, but you can view the currently configured secrets by running `config-secrets-show`. Updating the secrets happens through `config-secrets-edit`,

Storage

Wallypub stores your keys in your devices' keyring. You will note there are separate commands for showing the config and showing the secrets. This is partially to prevent printing your config for troubleshooting and accidentally exposing your sensitive values, but also because the secrets are not store in a plaintext settings file.

Troubleshooting

Wallypub generates a `debug.log` file which can be found in your ~/.local/share/Wallypub/ directory

If you want to see debug output in your terminal, set the `log_level` to `10` in your base settings.

Issues

There are two ways to get bugs addressed. Reach out to the jeremiah@micro.glasshoundcomputing.com account or follow the Bug Reports section of the CONTRIBUTING guidelines.

Frequently Asked Questions

Why the name?

Wallabag + Epub = Wallypub.

Doesn't Wallabag have some Ereader integration?

Yes! There are many. KOReader, for example, has great integration with Wallabag; however, I found that I wanted the news that makes it to my e-reader to be more curated and for the experience of reading articles to be more intentional.

How long will you support Wallypub?

I consider this to be a complete tool for the scope of the design, but it is very possible that users of this software have ideas that I want to implement or pull in through merge requests. While feature updates are possible, this project is being released in maintenance mode.