Newsman User Guide

Zeos-Ctrl

June 22, 2023

Contents

1 About

Newsman is a command line program that allows you to construct a mailing list and send out newsletters to the people on that list. It can work attached to the terminal or detached as a daemon and run as a background process!

2 Installation

Newsman can be installed from source for anyone familiar with that process, a database must also be constructed in order to complete the installation. The database schema can be found in the git repo under mailing_list.sql and MUST BE INSTALLED INTO A DATABASE BASED ON MYSQL because at the moment thats the only supported database type, i.e, mariadb. After installing the package a config file should have been created called .config/newsman/newsman.toml this file contains all the information needed by the program to work, this includes:

- url: This is the database url, for example mysql://root:password@localhost/emails
- dir: This is the directory the newsletters are stored in, it defaults to .config/newsman/newsletters/
- smtp_username: This is the username for the smtp client such as, example@mail.com.

- smtp_password: This is the password for the email.
- sender: Should be the same as smtp_username.
- relay: Your smtp relay for example, mail.example.com.

3 Usage

Newsman comes with multiple flag options which can be found with the newsman -h command:

- -a (Email) Adds an email to the mailing list.
- -r (Email) Removes an email from the mailing list.
- -j (Newsletter Name) Starts a mailing job for a specified newsletter.
- -t (Time) Time to delay the newsletter from being sent, defaults to 0s.
- -e (Execute) Executes all mailing jobs, given true or false.
- -d (Daemon) Runs the program as a daemon, given true or false.
- -debug Turns debugging information on.
- -h, -help (Help) Prints help.
- -V, -version (Version) Prints version.

4 Contributing

When contributing to this project make sure to follow these steps:

- Open an issue with your planned change
- Make a branch for your change
- Make your change following the projects structure
- Open a pull request to merge your branch

These steps ensure the software doesn't suffer from feature creep and keeps the codebase as clean as possible. When writing the code make sure it follows Rust best practices as well as being placed in an appropriate file. For example if you write a function for handling emails make sure it's in the emails.rs file. If adding a new type of function make a new file in the src directory called function.rs, where function is whatever your function relates to. Any tests should be written at the bottom of the that functions file.