

Redesign GNU Mailman Documentation Structure

Personal Information

- **Name:** Ananya Maiti
- **Telephone:** +91 890 025 3322
- **Email:** ananyoevo@gmail.com
- **Timezone:** UTC +5.30 (IST)
- **IRC handle:** ananyo on irc.freenode.net
- **Source control username (GitHub/GitLab):** ananyo2012
- **Instant Messaging information:**
 - Twitter: MaitiAnanyo
- **Blog:** <https://ilovefoss.wordpress.com/>
- **Employer:** Cadence Design System

Why I should be the chosen One

I believe that good documentation is one of the pillars of maintaining a widely used project. It is as important as writing tests for a project. And I try to enforce the same in every project I do. I am familiar with documentation tools/languages like markdown, Restructured text, Sphinx and blogging frameworks like jekyll, pelican. I am also proficient in git workflow and working collaboratively.

I am well aware that the project is to Restructure the Documentation so my primary aim would be to Chapter the existing documentation properly and organize them to the relevant sections. There is only one section related to Subscriber guide which will have new content for which I will take adequate review for adding the needed content.

I have been a GSoC 2016 student at publiclab.org and one of my tasks in the project was to make the doc newcomer friendly and create first timer issues. I have also been a mentor for GSoC 2017 at Publiclab and Google Code-in 2016 at FOSSASIA. So I am upfront with teamwork.

I also love Python and have been a regular attendee in Python meetups and PyCons in India. For my work I write my automation scripts in Python and also love to use python for my hobby projects. I am also an open source evangelist and keep on the lookout to contribute to open source projects.

Also I like to write about my experiences in my blog - <https://ilovefoss.wordpress.com/>

Resume/Link to previous work

Merged PR (mailman-suite-doc): https://gitlab.com/mailman/mailman-suite-doc/merge_requests/32

GSoC 2016 Project report:

<https://publiclab.org/notes/ananyo2012/08-20-2016/gsoc-2016-final-work-product-of-expanded-q-a-system-for-publiclab-org>

Blogs related to my GSoC 2016 project: <https://ilovefoss.wordpress.com/tag/gsoc/>

Recent research blog on python-requests:

<https://ilovefoss.wordpress.com/2018/08/19/understanding-python-requests/>

Resume link: https://drive.google.com/file/d/1WFLNyImE2FJZJhk64iVHwJ1O_Vgc0muM/view?usp=sharing

Project Proposal

Proposal Title - Redesign Document Structure for GNU Mailman

Proposal Abstract

The project aims to restructure the Documentation for GNU Mailman. The current documentation contains most of the information about the installation and setup and has a comprehensive guide from developers or site admins. But for subscribers or list admins there is very less documentation available.

Also there are 2 main places where the documentation is distributed -

<https://mailman.readthedocs.io/en/latest/> and <http://docs.mailman3.org/en/latest/> and individual documentation of Hyperkitty and postorius is also available. The effort will be to create a unified documentation for GNU Mailman suite that will serve as a guide for site admins, list owners, subscribers and developers, and refer to the other links for the project specific details.

Also a Troubleshooting section at the end will be helpful for site admins and developers working with Mailman.

With that in mind I have prepared an initial draft of the contents. Though this is not a final state of the work and is open for discussion.

Table of Contents

1. Overview of GNU Mailman

This chapter provides an overview of GNU Mailman project. This will mostly have the same content as the current content in GNU Mailman overview. The copyright, history and project details. A bonus would be adding a high level diagrammatic view of the project. Diagrams are good in explaining a project structure and tend to make it easier to understand.

2. Installation and Configuration

This chapter will contain all necessary information regarding installing and configuring Mailman in a production environment. The content will be taken from the existing mailman suite doc and GNU Mailman doc. The chapter should be a comprehensive guide to get the mailman suite running in production with all the necessary features and also let the reader understand the bells and whistles around Mailman. Following would be the sections under this chapter

a. Installing Mailman 3 suite (prodsetup)

- b. Setting up database
- c. Setting up Mail Server
- d. Configuring Mailman Core
- e. Configuring Web Frontend
- f. Mailman 3 standalone installation
- g. Postorius additional configuration
- h. Hyperkitty additional configuration
- i. Archivers
- j. Plugins
- k. Migrating from Mailman 2.1 to Mailman 3

3. Subscribers Guide and Mailing list usage

This chapter will provide an insight to the users/subscribers on how to use the full potential of a mailing list running with mailman. After reading this a subscriber should be able to effectively use the mailing list and also be able to set preferences required by the user. Also this will point out the “Easter eggs” in mailing list communication that a subscriber needs to know. Following would be the sections under this chapter

- a. Subscribing to a mailing list - (Include sign up options as well)
- b. Delivery options - Regular and digest emails
- c. Viewing list mails in popular mail clients
- d. Managing your Mailman account - Will include both account and Mailman settings
- e. Unsubscribing from a list
- f. Viewing Archives - Will include browsing options available in Hyperkitty UI. Explain options like
 - i. Filtering threads by date
 - ii. Filtering threads by Newest, Popularity, Activity, Post, Favorites
 - iii. Browsing threads
- g. Attachments and HTML content
- h. Hyperkitty interactive features
 - i. Upvote/Downvote
 - ii. Tags
 - iii. Replying from UI
 - iv. Start new thread from UI
 - v. Hyperkitty's format of displaying emails and threaded conversation
 - vi. Read/Unread messages
 - vii. Downloading archived emails
- i. Effective mailing list guidelines
 - i. Top posting/Bottom posting, Inline replies
 - ii. Text based emails vs HTML emails
 - iii. Inserting links

4. Managing lists using Mailman suite

This chapter will explain the various user roles present in Mailman, what privileges a user with a specific role has and what are the available list options. This will also be a guide to the Admins of the list to better manage lists. Following would be the section under this chapter

- a. Understanding user roles - Explain various privileges of user roles and hierarchy of privileges and fallback to higher-level configurations . Currently the following roles exist in Mailman
 - i. Owner
 - ii. Moderator
 - iii. Member
 - iv. Non-member
 - v. Admin (superuser)
 - vi. Site Admin
 - vii. Domain AdminWe can list the privileges in tabular form
- b. Site Admin Options
 - i. Creating a new list
- c. Domain Admin options
 - i. Adding a domain
 - ii. Adding/Editing template
- d. List Admin Options
 - i. Managing subscriptions (including mass operations)
 - ii. Ban/Unban users
 - iii. Managing Owner/Moderator privileges
 - iv. Header filters
 - v. List settings - Some of the options in these may need a bit of explanation e.g. DMARC

5. Troubleshooting

The [Site Admin tasks](#) and [List Admin tasks](#) wiki have a bunch of FAQs and Troubleshooting information. We can list the known issues and list them out here. Also we can look out for any new FAQs that come up in the way. A recent blog post that caught attention was a post regarding DMARC - <https://kushaldas.in/posts/dmarc-mailing-list-yahoo-and-gmail.html>

6. Developers guide to Mailman

This chapter will contain all the developer related information starting from Installation, architecture, the internals of Mailman to the contribution guidelines for developers wanting to contribute to Mailman. This will link to the Hyperkitty/Postorius documentation as well. Following will be the sections under this chapter

- a. Mailman suite architecture (Diagrammatic representation)
- b. Devsetup Instructions
- c. Mailman Core architecture
- d. Mailman REST API

- e. Postorius/Hyperkitty architecture (Link to relevant url)
- f. Mailman internals
 - i. Mailman Core Models
 - ii. Runners
 - iii. Chains
 - iv. Rules
 - v. Handlers
 - vi. Commands
 - vii. Core
 - viii. App
 - ix. List Styles
 - x. MTA
 - xi. Mailman Runner control
- g. Internationalization
- h. Contributing to Mailman 3 (We can add something on CI that mailman uses and to check for CI checks on raising a MR)
- i. GNU Mailman Coding Style Guide

7. Release Notes

This will have the same content as the present Release Notes. Just mentioning here to have it in a separate Chapter.

8. Miscellaneous

This will contain any Miscellaneous information related to Mailman. Some possible things that can be added from existing wiki/docs are

- Any OS specific issues
- Improving performance related information
- Mailman deb/rpm package info - State of these packages (If they are stale we can perhaps ask for contribution)

These are just my initial ideas. Will discuss on the same with developers and decide what can be added in this chapter

Timeline and Milestones

Community Bonding Period (August 1 - September 1)

I have already installed the mailman suite locally and also individual installations of hyperkitty and postorius. I am also familiar with Sphinx and I am able to build the docs using it. I am fairly familiar with Restructured text. The aim during Community bonding bonding period would be

- Fine tune the installation to be able to browse all features
- Get more familiar with writing docs using Restructured text
- Discuss more on the Documentation structure and come to a consensus on how to chapter the various topics. I intend to do this early on even before the Community bonding period so there is adequate time for this discussion. Will post an issue (with the Table of Contents) for this soon

- Keep in touch with mentors and community and communicate effectively

Task Freeze Timeline (August 24 - September 1)

I have created an issue for the Documentation Structure at <https://gitlab.com/mailman/mailman/issues/612> so that we can have adequate feedback regarding the same. During this period I will finalize the Tasks (and Milestones) that I am supposed to accomplish in my proposal during the GSoD timeline.

Milestone 1 : Chapter Restructure and Mailman Overview

Week 1 (September 2 - September 8)

- Reorganize sidebar and create Merge Request of first draft of Chapters after final discussion
- Integrate current available Overview documentation

Milestone Criteria:

- Merge Request for sidebar with the initial approved draft of the chapters and sections
- Linked Overview section

Deliverables:

- Updated sidebar with Reorganized Chapters from first draft
- Updated overview documentation

Milestone 2 : Mailman suite installation and Mail server setup

Week 2 (September 9 - September 15)

- Read Mailman suite installation documentation, verify the content and link it to Mailman docs
- Amend Mail server documentation. Here the aim would be to reorganize content a bit more. The doc already explains how to configure different mail servers. What is doesn't include is how to install or start/stop a server. Refer to the mail server docs and add this missing piece to the doc
- Merge Requests: Mailman suite installation changes, Mail server, Configuration amendments

Milestone Criteria:

- Merge Request for linking and updating mailman suite production installation doc
- Merge Request for updating Mail server doc

Deliverables:

- Mailman suite installation docs
- Updated Mail server docs

Milestone 3 : Web Front End, Plugin, Archiver configuration

Week 3 (September 16 - September 22)

- Link Configuring Web Front End documentation. Here the aim would be to do verify the content and add any changes if needed

- Read Plugins and Archivers doc and look for any amendments required. Add any pluggable interfaces if any.
- Link the Mailman 2.1 to Mailman 3 migration guide in this chapter. Migration itself is a separate project which will involve a lot of exploration and effort. So we will keep the existing content available.

Milestone Criteria:

- Merge request with Web Front End, Plugins and Archivers doc linked with necessary amendments
- Linked Migration doc

Deliverables:

- Reorder/Amend Plugin and Archiver doc
- Add Mailman 2.1 to 3 migration doc

Milestone 4 : Subscribers Guide

Week 4 (September 23 - September 29)

- Add Subscription/unsubscription docs
- Mailing list delivery modes
- Explore filtering options in popular mail clients and document the same
- Subscriber guide to Mailman settings

Milestone Criteria:

- Merge Request for adding content for Subscription/Unsubscription + Delivery mode + Mail clients and get it reviewed
- Merge request for Mailman settings for Subscribers and put it for review

Deliverables:

- Add Subscribers guide docs as mentioned above. I will try to add any subscriber information related to mail delivery functionalities/postorius web ui

Milestone 5: Hyperkitty Archive Feature and Mailing list guidelines

Week 5 (September 30 - October 6)

- Viewing and browsing list Archives in Hyperkitty
- How Hyperkitty handles HTML emails/ attachments
- Hyperkitty interactive features
- Downloading mail archives
- Using mailing lists effectively - A lot of these is inspired from <http://www.shakthimaan.com/downloads/glv/presentations/mailling-list-etiquette.pdf> . Though it largely describes mailing list etiquette a lot of information is relevant to effective usage of mailing lists
- Merge requests: Hyperkitty features, Effective mailing list usage guidelines

Milestone Criteria:

- Merge request for explaining archiving features of Hyperkitty. Get it reviewed by mentors
- Merge request for adding Effective Mailing list guidelines. Put it for review.

Deliverables:

- Add doc for Hypekitty archiving features
- Add Effective mailing list usage guide

Milestone 6: User Roles and privileges + Site and Domain Admin options

Week 6 (October 7 - October 13)

- Understand the various user roles and privileges
- Work with developers to create a table explaining user privileges
- Document Site Admin Options
- Document Domain Admin options

Milestone Criteria:

- Merge Request for User Roles and privileges Table. Put it for review
- Merge Request for Site Admin + Domain Admin options. Put it for review

Deliverables:

- Add Table explaining user privileges
- Add Site Admin + Domain Admin options

Milestone 7: Document List Admin options

Week 7 (October 14 - October 20)

Document various site Admin options like

- Managing subscription. Ban/Unban users
- Assigning someone as Owner/Moderator, header filters
- Document various list settings

Milestone Criteria:

- Merge request documenting various list Admin options (Subscription, Header Filters, Assigning ownership, List Settings doc)

Deliverables:

- Document various list admin options

Milestone 8: Mailman Suite Architecture

Week 8 (October 21 - October 27)

- Understand and create a high level diagram of mailman suite . How Mailman core interacts with postorius and Hyperkitty
- Test dev setup instructions and link them to the section
- Understand Mailman core architecture and prepare a rough sketch with help from developers
- Merge Requests: Mailman suite architecture, Devsetup instructions

Milestone Criteria:

- Merge Request for Mailman suite architecture
- Linked dev setup docs. Update content if required

Deliverables:

- Add Mailman suite architecture diagram
- Link Devsetup docs

Milestone 9: Mailman Core Architecture and Models

Week 9 (October 28 - November 3)

- Prepare a diagram for Mailman core architecture and get it reviewed
- Understand REST API docs and Models in Mailman core and check if any changes are needed

Milestone Criteria:

- Merge Request for Mailman Core architecture
- Merge Request for Model/API related changes if required

Deliverables:

- Mailman core architecture diagram
- Updated Models/api docs

Milestone 10: Mailman Internals and Localization

Week 10 (November 4 - November 10)

- Understand Mailman internals like Runners, Rules, Handles, Chains and see where it needs to be placed in the Developer documentation
- Localization doc seems to be missing in Internationalization. Add that and update the documentation after discussing with developers
- Update Mailman 3 contribution guidelines - Add a git tutorial link (I guess I saw it in a mailman wiki but the doc seems to be missing the link), Add something about CI mailman uses

Milestone Criteria:

- Merge Request with updated order of internals
- Merge Request for missing Localization doc
- Suggested Contribution changes if needed

Deliverables:

- Updated Mailman internals order
- Updated Localization and contribution guidelines

Milestone 11: Troubleshooting

Week 11 (November 11 - November 17)

Mailman already has a lot of docs regarding Troubleshooting in its wikis e.g. [Site Admin tasks](#) and [List Admin tasks](#) . I will use this week to discuss with the developers the issues they faced and list the known issues that can be added to the Troubleshooting doc

Milestone Criteria - Add about 3-5 Troubleshooting points and create a Merge Request

Deliverables - Add Troubleshooting doc

Milestone 12: Miscellaneous Docs

Week 12 (November 18 - November 24)

Any other additional content can be added in the miscellaneous section. Will discuss with mentors and developers to understand what type of things can be added in this chapter and proceed as per discussion.

Milestone criteria - Add 3-5 points in this chapter and create a Merge Request

Deliverables - Add Miscellaneous info docs

Final Week (November 25 - November 29)

I will use this week to complete any pending tasks or go over Merge requests if any. Also I will use this week to compile the final project report documentation required for evaluation.

Availability/Schedule

I am working full time as a Software Engineer, my work week is from Mon-Fri (weekdays) and usual timings is 9 am - 5.30 pm. On weekdays I will be able to work in the evenings and devote about 3-4 hrs per day. On weekends I can work on about 8-10 hours including morning and evening.

If there is any change in schedule/vacation I will keep my mentors informed in advance.