Mailman Suite Documentation

Release 3.0

Free Software Foundation

Jun 12, 2017

Contents

The Mailman home page is http://www.list.org, and there is a community driven wiki at http://wiki.list.org.

Mailman 3.1 was released on May 25, 2017. The Mailman Suite consists of 5 individual projects Below are links to documentation for each of the projects.

Those packages are copyrighted by the Free Software Foundation and distributed under the terms of the GNU General Public License (GPL) version 3 or later.

- Mailman Core - the mailing list manager core (required)
- Postorius - the adminstrative web user interface
- MailmanClient - the official REST API Python bindings
- HyperKitty - the web archiver
- HyperKitty Mailman plugin - archiver plugin for Core

CHAPTER 1

The Installation Guide

Mailman Suite Installation

Mailman 3 consists of a collection of separate-but-linked projects, each of which has its own setup guide. This is handy when you want to focus on development for a single piece of Mailman, but is less useful for setting up a complete installation suitable for production use.

Setting up Web UI

Postorius and Hyperkitty are Mailman's official Web UI and Archiver. Both of them are Django based apps and can be integrated into an existing Django installation/website. Here are the parameters that will affect how they function. These parameters are configured in your Django Site's `settings.py`.

- `MAILMAN_REST_API_URL` : Complete URL to the Core's REST API Server. Usually, Mailman Core listens on port 8001 for REST API. e.g. `http://localhost:8001/`
- `MAILMAN_REST_API_USER` : Username for the Core's REST API, default value in core is 'restadmin' if not set.
- `MAILMAN_REST_API_PASS` : Password for Mailman REST API User, default value in core is 'restpass' if not set.
- `MAILMAN_ARCHIVER_KEY` : The Key used to authenticate the emails for archiving in Hyperkitty. Its value should be exactly same as set in Core.

  Also note that the value in `settings.py` will be within single quotes, but in `mailman.cfg` it will be without any quotes.
- `FILTER_VHOST` : Filter the list of available lists in Hyperkitty depending on the domain it is being currently served from. Mailman 3 supports multiple domains in a single installation.
- Add `compressor.finders.CompressorFinder` to your `STATICFILES_FINDERS`.
- `LOGIN_URL` = 'account_login'
- `LOGIN_REDIRECT_URL` = 'list_index'

- LOGOUT_URL = 'account_logout'
- This setting is for django-compressor which is used here to compile and compress static files:

```
COMPRESS_PRECOMPILERS = (
  ('text/x-scss', 'sassc -t compressed {infile} {outfile}'),
  ('text/x-sass', 'sassc -t compressed {infile} {outfile}'),
)
```

- Hyperkitty uses **'django-haystack'_** for its full text search:

```
HAYSTACK_CONNECTIONS = {
      'default': {
      'ENGINE': 'haystack.backends.whoosh_backend.WhooshEngine',
      'PATH': os.path.join(BASE_DIR, "fulltext_index"),
      # You can also use the Xapian engine, it's faster and more accurate,
      # but requires another library.
      # http://django-haystack.readthedocs.io/en/v2.4.1/installing_search_engines.
↪html#xapian
      # Example configuration for Xapian:
      #'ENGINE': 'xapian_backend.XapianEngine'
      },
}
```

- Hyperkitty uses **'django_q'_** as a task queue. It supports various different backends like Redis, Disque, IronMQ, SQS etc. Please check the documentation to better understand how to configure it. The most basic setup where it uses Django orm as the queue can be configured using the settings below:

```
Q_CLUSTER = {
   'timeout': 300,
   'save_limit': 100,
   'orm': 'default',
}
```

These settings below are required by Django-Allauth:

- ACCOUNT_AUTHENTICATION_METHOD = "username_email"
- ACCOUNT_EMAIL_REQUIRED = True
- ACCOUNT_EMAIL_VERIFICATION = "mandatory"
- ACCOUNT_DEFAULT_HTTP_PROTOCOL = "http"
- ACCOUNT_UNIQUE_EMAIL = True

Setting up mail server

In order to run Mailman you need have a mail transport agent (MTA), or mail server, setup on your server. It should be able to relay emails to and from Mailman. Both Mailman Core and your Django server should be configured to send emails through your MTA.

Mailman Core

Mailman Core handles all the tasks related to sending and receiving emails for the mailing lists. It supports various different MTAs.

- **'Postfix'_**: To setup Postfix, add the following configuration to master.cf:

```
# Support the default VERP delimiter.
recipient_delimiter = +
unknown_local_recipient_reject_code = 550
owner_request_special = no
transport_maps =
    hash:/path-to-mailman/var/data/postfix_lmtp
local_recipient_maps =
    hash:/path-to-mailman/var/data/postfix_lmtp
relay_domains =
    hash:/path-to-mailman/var/data/postfix_domains
```

Mailman’s `var` directory can vary according your source of installation. Please refer to the documentation provide with your source or ask Mailman Developers at mailman-developers@python.org.

To configure Core to use Postfix, add the following configuration to your `mailman.cfg` configuration, be sure to replace `mail.example.com` with your email domain:

```
[mta]
incoming: mailman.mta.postfix.LMTP
outgoing: mailman.mta.deliver.deliver
lmtp_host: mail.example.com
lmtp_port: 8024
smtp_host: mail.example.com
smtp_port: 25
```

- **‘Exim4‘_**: To setup Exim4, add the following files to your configuration as file names mentioned above:

```
# /etc/exim4/conf.d/main/25_mm3_macros
# The colon-separated list of domains served by Mailman.
domainlist mm_domains=list.example.net

MM3_LMTP_PORT=8024

# MM3_HOME must be set to Mailman's var directory, wherever it is
# according to your installation.
MM3_HOME=/path-to-mailman/var/
MM3_UID=list
MM3_GID=list

################################################################
# The configuration below is boilerplate:
# you should not need to change it.

# The path to the list receipt (used as the required file when
# matching list addresses)
MM3_LISTCHK=MM3_HOME/lists/${local_part}.${domain}

# /etc/exim4/conf.d/router/455_mm3_router
mailman3_router:
driver = accept
domains = +mm_domains
require_files = MM3_LISTCHK
local_part_suffix_optional
local_part_suffix = \
-bounces   : -bounces+* : \
-confirm   : -confirm+* : \
-join      : -leave     : \
-owner     : -request   : \
-subscribe : -unsubscribe
transport = mailman3_transport

# /etc/exim4/conf.d/transport/55_mm3_transport
mailman3_transport:
driver = smtp
protocol = lmtp
allow_localhost
hosts = localhost
port = MM3_LMTP_PORT
rcpt_include_affixes = true
```

You should note that Exim4 configuration above doesn't support multiple domains like Postfix does. Please change the variables in the configuration above before installing it.

To configure Mailman to use Exim4 add the following to your `mailman.cfg`

```
[mta]
incoming: mailman.mta.exim4.LMTP
outgoing: mailman.mta.deliver.deliver
lmtp_host: mail.example.com
smtp_host: mail.example.com
lmtp_port: 8024
smtp_port: 25
```

- Mailman also supports **'Sendmail'_** and **'qmail'_**. Please check the Mailman Core documentation for **'sendmail setup'_** and **'qmail setup'_** to configure them.

Postorius & Hyperkitty

If your Django site is configured to send emails, there is no additional setup required for Postorius and Hyperkitty. Here are some settings that determine how your emails from them look like:

- `DEFAULT_FROM_EMAIL` : This is the default address that used used as the `FROM` header in all the emails from your django installation.
- `SERVER_EMAIL` : This is the address from which the errors emails will be sent to you.

Note that both of these are general Django related settings and will affect other parts of your Django installation too!

Setting up a web server

These instructions are to setup your Django website behind a webserver. We are using a uwsgi as the wsgi server to communicate between the webserver and Django. First setup uwsgi using the following configuration file:

```
[uwsgi]
# Port on which uwsgi will be listening.
http = :8000

# Move to the directory wher the django files are.
chdir = /path-to-django-directory/

# Use the wsgi file provided with the django project.
wsgi-file = wsgi.py

# Setup default number of processes and threads per process.
```

```
master = true
process = 2
threads = 2

# Drop privielges and don't run as root.
uid = 1000
gid = 1000

# Setup the django_q related worker processes.
attach-daemon = ./manage.py qcluster

# Setup the request log.
req-logger = file:/path-to-logs/logs/uwsgi.log

# Log cron seperately.
logger = cron file:/path-to-logs/logs/uwsgi-cron.log
log-route = cron uwsgi-cron

# Log qcluster commands seperately.
logger = qcluster file:/path-to-logs/logs/uwsgi-qcluster.log
log-route = qcluster uwsgi-daemons

# Last log and it logs the rest of the stuff.
logger = file:/path-to-logs/logs/uwsgi-error.log
```

Note that in the above configuration, there is a command called `python manage.py qcluster` which run the `django-q` processes. You can remove this from here if you want to manage this yourself via some other init process.

Have a look at **‘uwsgi‘_** documentation to learn more about different configuration options. There are 1000s of them.

Nginx Configuration

You can reverse proxy the requests to uwsgi server using Nginx. Uwsgi has a special protocol called uwsgi protocol that is available in Nginx as a plugin. Add the following configuration to your sites-availbale in Nginx:

```
server {

      listen 443 ssl default_server;
      listen [::]:443 ssl default_server;

      server_name MY_SERVER_NAME;
      location /static/ {
           alias /path-to-djang-staticdir/static/;
      }
      ssl_certificate /path-to-ssl-certs/cert.pem;
      ssl_certificate_key /path-to-ssl-certs/privkey.pem;

      location / {
              # First attempt to serve request as file, then
                  include uwsgi_params;
              uwsgi_pass 172.19.199.3:8000;

      }

}
```

Fill in the appropriate paths above in the configuration before using it.

CHAPTER 2

The User Guide

GNU Mailman 3.1 - List Member Manual

Mailman is free software for managing electronic mail discussion and e-newsletter lists. Mailman is integrated with the web, making it easy for users to manage their accounts and for list owners to administer their lists. Mailman supports built-in archiving, automatic bounce processing, content filtering, digest delivery, spam filters, and more.

This guide contains instructions for members of Mailman mailing lists so they can learn to use the features available to them. This focuses on the web interface and includes sections on joining and leaving lists, editing options and other subscriber-level tasks.

This guide is written for Mailman 3.1. If you are using Mailman 2.1 (our popular previous stable release), you should see the Mailman 2.1 Members Manual.

Introduction to Mailman Suite

It may be easier to think of Mailman 3 as a single piece of software that does mailing list management, but it's actually a set of interconnected pieces of software under the hood.

The big pieces you care about as a user are as follows:

- **Mailman Core** - This is the "core" of Mailman that handles getting and sending email and stores all your email-related preferences.
- **Postorius** - This is the web interface to Mailman, which allows users to subscribe and unsubscribe from mailing lists and set preferences from the web.
- **Hyperkitty** - This is the archiver for Mailman, which allows users to view and interact with list archives from the web.

These were divided up so that you could replace pieces of Mailman as needed. For example, if a site already had a user settings page, you might want to run a modified version of Postorius so that users can set all their preferences in one place, or you might want to replace the web interface entirely.

This document is going to assume you're using these pieces together in their default states. If your Mailman installation is customized or has replaced any of these pieces, you'll need to adjust the instructions accordingly.

Mailing List Terminology

- A "**post**" typically denotes a message sent to a mailing list.
- People who are part of an electronic mailing list are usually called the list's "**members**" or "**subscribers**."
- "**List administrators**" are the people in charge of maintaining that one list. Lists may have one or more administrators.
- A list may also have people in charge of reading posts and deciding if they should be sent on to all subscribers. These people are called "**list moderators**."
- Often more than one electronic mailing list will be run using the same piece of software. The person who maintains the software which runs the lists is called the "**site administrator**." Often the same person who acts as site administrator also administrates individual lists.

Translating from our examples to real lists

Often, it's easier to simply give an example than explain exactly how to find the address for your specific list. As such, we'll frequently give examples for a fictional list called LISTNAME@DOMAIN whose list information page can be found at http://WEBSERVER/mailman3/lists/LISTNAME.DOMAIN.

Neither of these are real addresses, but they show the form of a typical list address. The capital letters used for the list-specific parts of each address should make it easier to see what should be changed for each list. Although specific list configurations may be different, you will probably be able to just replace the words given in capital letters with the appropriate values for a real list:

LISTNAME The name of your list.

DOMAIN The name of the mail server which handles that list.

WEBSERVER The name of the web server which handles the list web interface. This may be the same as DOMAIN, and often refers to the same machine, but does not have to be identical.

As a real-life example, if you are interested in the mailman-users list that runs on Mailman3.org, you'd make the following substitutions: LISTNAME=mailman-users, DOMAIN=mailman3.org, WEBSERVER=lists.mailman3.org. As such, for the mailman-users mailing list on mailman3.org, the list information page can be found at the URL http://lists.mailman3.org/mailman3/lists/mailman-users.mailman3.org/. (These, unlike most of the examples given in this document, are real addresses.)

Most lists will have this information stored in the List-* headers. Many mail programs will hide these by default, so you may have to choose to view full headers before you can see these informational headers.

I need to talk to a human!

If you have any trouble with any of these commands, you can always reach the person or people in charge of a list by using the list administrator email address. The list administrators can help you figure out how to do something, subscribe/unsubscribe you, or change your settings if you are unable to change them yourself for some reason. Please remember that many mailing list administrators are volunteers who are donating their spare time to run the list, and they may be very busy people.

This list administrator email address is in the form LISTNAME-owner@DOMAIN, where LISTNAME is the name of the list and DOMAIN is the name of the server. So for an example list called <wolfhounds@example.com> the administrators could be reached using <wolfhounds-owner@example.com>

This email address can also be found on the list information pages.

Making a Mailman account

In order to manage your options and easily subscribe to or unsubscribe from Mailman lists, you typically want to make an account. There is a "sign up" link on most list pages (usually displayed in the upper right of the page), or you can go directly to the sign up interface at a URL that will be something like http://WEBSERVER/accounts/signup/

If you've been subscribed to a list without making an account (because you did this yourself or because your lists were migrated from a Mailman 2.1 setup) you can make an account using the same email address and once you've confirmed that you have access to that email, you'll be able to edit all the associated options.

Note that in Mailman 3, you can actually have multiple email addresses associated to the same user account, so you don't need to make many separate accounts to handle your permissions.

Subscribing and Unsubscribing

How do I join a list?

1. Go to the list information page for the list you want to join. This will be something like http://WEBSERVER/mailman3/lists/LISTNAME.DOMAIN
2. Usually, it is a good idea to make an account first using the "sign up" option (on the upper right). This account will allow you to change your settings later and make it easier for you to unsubscribe.
3. Once you are signed in, go back to the list information page and there will be a large section labelled "Subscribe to this list" where you can choose the email address you want to use and optionally choose a display name. Fill out this form and click the "subscribe" button.
4. Alternatively, you can also join a list without signing in by using the subscribe boxes at the bottom of the list information page. If you need to edit your settings later, you will need to create an account associated with the same email address.

How do I leave a list?

1. Go to the list information page for the list you want to leave. This will be something like http://WEBSERVER/mailman3/lists/LISTNAME.DOMAIN
2. Log in to confirm that you are the owner of the address that you wish to unsubscribe. If you don't already have an account associated with that address, you may need to make one to prove that you are the correct owner of that email address.
3. Once you are logged in, there will be a section marked "Subscription / Unsubscription" that shows you the address which is subscribed to that list and a large "unsubscribe" button you can click to leave the list.

Changing your list settings

Mailman has a number of different settings for list subscribers as follows:

Delivery status Set this option to Enabled to receive messages posted to this mailing list. Set it to Disabled if you want to stay subscribed, but don't want mail delivered to you for a while (e.g. you're going on vacation). If you disable mail delivery, don't forget to re-enable it when you come back; it will not be automatically re-enabled.

Delivery mode If you select summary digests , you'll get posts bundled together (usually one per day but possibly more on busy lists), instead of singly when they're sent. Your mail reader may or may not support MIME digests. In general MIME digests are preferred, but if you have a problem reading them, select plain text digests.

Receive own postings Ordinarily, you will get a copy of every message you post to the list. If you don't want to receive this copy, set this option to No. Note that some mail services (most prominently Gmail) will suppress this copy no matter what you do. If you need to know when your email went through and your mail provider is blocking or removing the copy, you can also use the "Acknowledge posts" option to get a separate email acknowledging your post.

Acknowledge posts Receive acknowledgement mail when you send mail to the list? The options are yes or no. This is useful if your mail provider is making it difficult for you to know if your post has gone through.

Hide address When someone views the list membership, your email address is normally shown (in an obscured fashion to thwart spam harvesters). If you do not want your email address to show up on this membership roster at all, select Yes for this option.

Avoid Duplicates When you are listed explicitly in the To: or Cc: headers of a list message, you can opt to not receive another copy from the mailing list. Select Yes to avoid receiving copies from the mailing list; select No to receive copies.

Each of these settings can be set globally, per address, or per list. Your per-list settings over-ride the per-address settings which over-ride the global settings.

How do I view my list settings?

To change any settings, you can go to your settings page.

1. Log in to Mailman.
2. Click on the dropdown menu by your username (in the upper right) and select "Mailman settings"
3. Alternatively, the URL for this page will be something like http://WEBSERVER/mailman3/accounts/subscriptions/LISTNAME.DOMAIN

How do I disable/enable my mail delivery?

You may wish to temporarily stop getting messages from the list without having to unsubscribe. If you disable mail delivery, you will no longer receive messages, but will still be a subscriber and will retain your other settings.

To disable or enable mail delivery from the web interface:

1. Log in and go to your list settings page
2. There is an option labelled "Delivery status" which you can enable or disable on your preferences tabs, either globally, per address, or per list subscription.

This can be handy in a many different cases. For example, you could be going on vacation or need a break from the list because you're too busy to read any extra mail. Many mailing lists also allow only subscribers to post to the list, so if you commonly send mail from more than one address (e.g., one address for at home and another for when you're travelling), you may want to have more than one subscribed account, but have only one of them actually receive mail. You can also use this as a way to read private archives even on a list which may be too busy for you to have sent directly to your mailbox. All you need to do is subscribe, disable mail delivery, and use your password and email to log in to the archives.

How can I start or stop getting the list posts grouped into one big email?

Groups of posts are called “digests” in Mailman. Rather than get messages one at a time, you can get messages grouped together. On a moderately busy list, this typically means you get one email per day, although it may be more or less frequent depending upon the list.

To change your digest settings:

1. Log in and go to your list settings page
2. There is an option labelled “Delivery Mode” which you can set on your preferences tabs, either globally, per address, or per list subscription.

There are a number of different options:

Regular You get an email every time the list sends one out.

Mime Digests MIME is short for Multipurpose Internet Mail Extensions. It is used to send things by email which are not necessarily simple plain text. (For example, MIME would be used if you were sending a picture of your dog to a friend.) A MIME digest has each message as an attachment inside the message, along with a summary table of contents.

Plain Text Digests A plain text digest is a simpler form of digest, which should be readable even in mail readers which don’t support MIME. The messages are simply put one after the other into one large text message.

Most modern mail programs do support MIME, so you only need to choose plain text digests if you are having trouble reading the MIME ones.

How do I stop or start getting copies of my own posts?

By default in Mailman, you get a copy of every post you send to the list. Some people like this since it lets them know when the post has gone through and means they have a copy of their own words with the rest of a discussion, but others don’t want to bother downloading copies of their own posts.

To receive or stop receiving your own posts:

1. Log in and go to your list settings page
2. There is an option labelled “Receive own postings” which you can set to yes or no on your preferences tabs, either globally, per address, or per list subscription.

Note: This option has no effect if you are receiving digests.

Despite the availability of this option, some mail hosts (such as Gmail) will hide these posts from you. You may wish to see “How can I get Mailman to tell me when my post has been received by the list?” as an alternative solution if your posts are being eaten by your mail host.

How can I get Mailman to tell me when my post has been received by the list?

On most lists, you will simply receive a copy of your mail when it has gone through the list software, but if this is disabled, your list mail delivery is disabled, you use a mail host such as Gmail which blocks copies of your post from being received, or you simply want an extra acknowledgement from the system, this option may be useful to you.

To receive or stop receiving your own posts:

1. Log in and go to your list settings page
2. There is an option labelled “Acknowledge Posts” which you can set to yes or no on your preferences tabs, either globally, per address, or per list subscription.

How can I hide my email address on the subscriber list?

When someone views the list membership, your email address is normally shown (in an obscured fashion to thwart spam harvesters), but you can hide the address if you want:

1. Log in and go to your list settings page
2. There is an option labelled "Hide adresss" which you can set to yes or no on your preferences tabs, either globally, per address, or per list subscription.

Note that this does NOT hide your address in the list archives (if the list has archives) or when it's sent out in emails, so a dedicated spammer could probably get your address in other ways.

How can I avoid getting duplicate messages?

Mailman can't completely stop you from getting duplicate messages, but it can help. One common reason people get multiple copies of a mail is that the sender has used a "group reply" function to send mail to both the list and some number of individuals. If you want to avoid getting these messages, Mailman can be set to check and see if you are in the To: or CC: lines of the message. If your address appears there, then Mailman can be told not to deliver another copy to you.

To avoid duplicates:

1. Log in and go to your list settings page
2. There is an option labelled "Avoid Duplicates" which you can set to yes or no on your preferences tabs, either globally, per address, or per list subscription.

Documentation for Mailman 3 List Owners and Site Administrators is not yet complete.

CHAPTER 3

The Contributor Guide

The Contributor Guide

Mailman 3 consists of a collection of separate-but-linked projects, each of which has its own development setup guide. This makes sense when you want to focus on a single piece of Mailman, but can make setting up the entire Mailman Suite in one go somewhat confusing. This guide attempts to move all the information currently in the wiki and various package documentation into a single "definitive" guide.

This document currently collates information from the following sources: 1. The Mailman Wiki:

- Main development guide: https://wiki.list.org/DEV/SetupDevEnvironment
- Hyperkitty development guide: https://wiki.list.org/HyperKitty/DevelopmentSetupGuide

2. Main package documentation on Readthedocs.io:

- Mailman core start guide: https://mailman.readthedocs.io/en/release-3.0/src/mailman/docs/START.html
- Mailman core "web ui in 5" guide: https://mailman.readthedocs.io/en/release-3.0/src/mailman/docs/WebUIin5.html
- Mailman core "archive in 5": https://mailman.readthedocs.io/en/release-3.0/src/mailman/docs/ArchiveUIin5.html
- Postorius dev guide: http://postorius.readthedocs.io/en/latest/development.html
- Hyperkitty dev guide: http://hyperkitty.readthedocs.io/en/latest/development.html

Getting prerequisites

For the most part, setup for each project will download any needed packages. However, you will need a few system packages to be sure you've got the necessary version of Python and its tools, git (to get the source code), postfix (a mail server), and a few other tools that are used during setup.

On Fedora, you probably want to run:

```
$ sudo yum install python-setuptools python-devel python-virtualenv python3-devel git␣
↪gcc nodejs-less postfix
```

On Debian and Ubuntu, this may be something like:

```
$ sudo apt-get install python-setuptools python-dev python-virtualenv python3-dev git␣
↪gcc node-less nodejs postfix
```

If you prefer, you can substitute Exim4 for Postfix. Postfix is the MTA used by most Mailman developers, but we do support Exim 4. (Sendmail support is very much desired, but the Mailman core developers need contributors with Sendmail expertise to help.)

You will also want tox to run tests. You can get this using "pip install tox".

HyperKitty also needs sassc. FIXME: add instructions on how to get sassc on a few platforms.

Set up a directory

Setting up the whole Mailman suite means you have to pull code from a bunch of different related repositories. You can put all the code anywhere you want, but you might want to set up a directory to keep all the pieces of mailman together. For example:

```
$ mkdir ~/mailman
# cd ~/mailman
```

For the rest of this development guide, we are going to assume you're using `~/mailman` as your directory, but you can use whatever you want.

Set up virtual environments

Mailman core currently requires Python 3, while Postorius and HyperKitty require Python 2.7. The easiest way to handle this is to make sure you have virtualenvs for both environments.

To create the virtualenvs, you'll use different commands for Python 3 and Python 2.7 as follows:

```
$ python3.5 -m venv venv-3.5
$ virtualenv venv-2.7
```

To activate a virtualenv, you need to run the appropriate activate script:

```
$ source venv-2.7/bin/activate
```

You *must* use `source` (or `.` if your shell is a pure POSIX shell). Make sure to always be using one of the virtualenvs when you're setting up and running Mailman.

Set up and run Mailman Core

First, get the code:

```
$ cd ~/mailman
$ git clone https://gitlab.com/mailman/mailman.git
```

To set up Mailman Core, you'll need to switch to your Python 3 virtualenv:

```
$ source venv-3.5/bin/activate
```

(Change the version number if you're using a different version of Python 3, currently 3.4, 3.5, and 3.6 are supported.)

Then, go into the mailman directory, run setup, and then run `mailman info` to be sure everything is set up correctly, and that the right settings are in place:

```
$ cd mailman
$ python setup.py develop
$ mailman info
```

You can edit your `mailman.cfg` file to make any necessary changes. Then start things up:

```
$ mailman start
$ cd ..
```

You may want to deactivate your virtualenv now, since you'll be using a different one for other components:

```
$ deactivate
```

Note that mailman just makes a var/ directory wherever you start it and uses that to store your data. This is great for the purposes of testing so you can easily make fresh installs, but might be confusing if you restart your instance later from a different location and don't have your original mailman.db file, or if you start looking around and finding var/ directories everywhere.

Later on, if you need to restart Mailman (i.e. if you get the error "Mailman REST API not available. Please start Mailman core.") then you can also do that by calling the `mailman` executable from the venv as follows:

```
$ ~/mailman/venv-3.5/bin/mailman start
```

Note that the `mailman` executable has several sub-commands. One that is particularly useful for debugging is `mailman shell`.

Set up Mailman Client

Get the code:

```
$ cd ~/mailman
$ git clone https://gitlab.com/mailman/mailmanclient.git
```

Then set up mailmanclient:

```
$ source venv-2.7/bin/activate
$ cd mailmanclient
$ python setup.py develop
$ cd ..
$ deactivate
```

Note that we're using the Python 2.7 virtualenv here. (`mailmanclient` can also be run in the Python 3 virtualenv for experimentation. For supported versions of Python, you don't need to be too careful about which interpreter you use because Python 2 and Python 3 do not share `.pyc` files.)

Set up Django-mailman3

This package holds the Django libraries and templates used by Postorius and HyperKitty.

Get the code and set it up:

```
$ cd ~/mailman
$ git clone https://gitlab.com/mailman/django-mailman3.git
$ source venv-2.7/bin/activate
$ cd django-mailman3
$ python setup.py develop
$ cd ..
$ deactivate
```

Set up and run Postorius

The Postorius documentation, including a more extensive setup guide, can be found here: http://postorius.readthedocs.org/

Make sure to install mailmanclient and django-mailman3 before setting up Postorius. (If you're following this guide in order, you've just done that.)

Get the code and run setup. Make sure you're in the 2.7 venv for Postorius:

```
$ cd ~/mailman
$ git clone https://gitlab.com/mailman/postorius.git
$ source venv-2.7/bin/activate
$ cd postorius
$ python setup.py develop
$ cd ..
$ deactivate
```

Postorius and HyperKitty both come with `example_project` directories with basic configuration so you can try them out. For this tutorial, however, we'll be using a project that combines both instead.

Set up a mail server

To be able to actually receive emails, you need to setup a mail server. Mailman core receives emails over LMTP Protocol, which most of the modern MTAs support. However, setup instructions are provided only for Postfix, Exim4 and qmail. Please refer to the MTA documentation at Mailman Core for the details.

You will also have to add some settings to your django configuration. The setup instructions are provided in django's email documentation.

Set up and run HyperKitty

Complete guide here: https://hyperkitty.readthedocs.org/en/latest/development.html

Make sure to install mailmanclient and django-mailman3 before setting up Hyperkitty. (If you're following this guide in order, you've just done that.)

Get the code and run setup:

```
$ cd ~/mailman
$ git clone https://gitlab.com/mailman/hyperkitty.git
$ source venv-2.7/bin/activate
$ cd hyperkitty
$ python setup.py develop
$ cd ..
$ deactivate
```

Postorius and HyperKitty both come with example_project directories with basic configuration so you can try them out. By default, they both use port 8000, so if you do want to run both example projects at the same time, do remember that you'll need to specify a different port on the command line for one of them.

However, we're going to run them both in a single Django instance at the end of this guide, so don't worry about ports right now.

Set up mailman-hyperkitty

mailman-hyperkitty is the package that actually sends the incoming emails to HyperKitty for archiving. Note that this is one of the components that uses Python 3.

Get it and set it up:

```
$ cd ~/mailman
$ git clone https://gitlab.com/mailman/mailman-hyperkitty.git
$ source venv-3.5/bin/activate
$ cd mailman-hyperkitty
$ python setup.py develop
$ cd ..
$ deactivate
```

You'll need to fix the default mailman-hyperkitty.cfg file to use the correct url for HyperKitty. If you're running it on http://localhost:8002 then you need to change base_url to match that.

Link Mailman to HyperKitty

Now you have to enable HyperKitty in Mailman. To do that, edit the mailman.cfg` (in ~/mailman/mailman/var/etc, or wherever the output of mailman info says it is) and add the following config. Note that you need to fill in the absolute path to your mailman-hyperkitty.cfg in the configuration below:

```
# mailman.cfg
[archiver.hyperkitty]
class: mailman_hyperkitty.Archiver
enable: yes
configuration: <absolute path to mailman-hyperkitty.cfg>
```

Run the Mailman Suite (combined hyperkitty+postorius)

You can run HyperKitty and Postorius as separate applications, but many developers are going to want to run them on a single server. The configuration files for this are in a repository called mailman-suite.

The first time you run the suite, you will want to set up a superuser account. This is the account you will use in the web interface to set up your first domains. Please enter an email address otherwise the database won't be setup correctly and you will run into errors later:

```
$ cd ~/mailman
$ git clone https://gitlab.com/mailman/mailman-suite.git
$ source venv-2.7/bin/activate
$ cd mailman-suite/mailman-suite_project
$ python manage.py migrate
$ python manage.py createsuperuser
```

You'll want to run the following commands in a window where you can leave them running, since it dumps all the django logs to the console:

```
$ python manage.py runserver
```

At this point, you should be able to see Mailman Suite running! In the default setup, you can go to http://127.0.0.1:8000 and start poking around. You should be able to use the superuser account you created to log in and create a domain and then some lists.

The default config file uses a dummy email backend created by this line in settings.py:

```
EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
```

Using this backend, all emails will be printed to the Postorius console (rather than sent as email) so you can get the url to verify your email from the console.

Don't leave the console email backend configured and running once you get to the point where you want to send real emails, though!