A modern Mutt Setup

- A modern mutt setup

A modern mutt setup

Original seen and Highly inspired by:

https://webgefrickel.de/blog/a-modern-mutt-setup

As the time of writing I am using a Debian 10 system, you maybe have to adopt some settings depending on your Mailprovider and/or Distribution. I wrote to the the Original Autor for approval, he used MacOS and I adopted it to fit with latest stable version of Debian.

You can find the newest Dotfiles here

https://git.tinfoil-hat.net/?p=dotfiles.git;a=summary

install dependencies

These are the main component's you'd need to install for a functional mailclient.

```
sudo apt-get install neomutt msmtp isync maildir-utils vim
```

But I am going to install a few other tools that provide additional features, like calendar, contacts, downloading attacments, search and display content:

```
sudo apt-get install vdirsyncer khard khal ripmime urlscan gnupg gpg-agent
```

After that we create the neccessary folders:

```
mkdir ~/Contacts
mkdir ~/Mail
mkdir ~/Mail/mail
mkdir ~/Mail/private
```

Note: *I had to store my neomutt config files in `~/.config/neomutt` in order to make neomutt recognze them. It wasn't enough to put them in `~/.mutt*`

Store your Credentials

To Store the Password of your email addresses, execute after you typed your Password, exit with ctrl+d

```
gpg --encrypt -o ~/.mutt/msmtp-mail.gpg -r mail@tinfoil-hat.net
gpg --encrypt -o ~/.mutt/msmtp-private.gpg -r mail@tinfoil-hat.net
```

See this comment. Also the GPG Agent will only cache one Key at the time. so when you want to send an email with the wrong key after using one other, the agent will not be able to use the right key.

Configure isync

Then we create the `~/.mbsyncrc` and edit the Options, here's an Example

```
################################
######## Account mail ##########
################################

IMAPAccount mail
Host mx.nchristian.net
port 143
User mail@tinfoil-hat.net
PassCmd "echo ${PASSWORD:-$(gpg --no-tty -qd ~/.config/neomutt/msmtp-mail.gpg)}"
SSLType STARTTLS
SSLVersions TLSv1.2
# ca-file location for Debian 10
CertificateFile /etc/ssl/certs/ca-certificates.crt
```

```
# Remote storage
IMAPStore mail-remote
Account mail

# Local storage
MaildirStore mail-local
Path ~/Mail/mail/
Inbox ~/Mail/mail/INBOX

Channel mail-inbox
Master :mail-remote:"INBOX"
Slave :mail-local:INBOX
Create Both
Expunge Both

Channel mail-drafts
Master :mail-remote:"Drafts"
Slave :mail-local:Drafts
Create Both
Expunge Both

Channel mail-sent
Master :mail-remote:"Sent"
Slave :mail-local:sent
Create Both
Expunge Both

Channel mail-trash
Master :mail-remote:"Trash"
Slave :mail-local:Trash
Create Both
Expunge Both

Channel mail-junk
Master :mail-remote:"Junk"
Slave :mail-local:Junk
Create Both
Expunge Both
```

```
Group mail
Channel mail-inbox
Channel mail-drafts
Channel mail-sent
Channel mail-trash
Channel mail-junk

################################
######## Account private ########
################################

IMAPAccount private
Host mx.nchristian.net
port 143
User @nchristian.net
PassCmd "echo ${PASSWORD:-$(gpg --no-tty -qd ~/.config/neomutt/msmtp-private.gpg)}"
SSLType STARTTLS
SSLVersions TLSv1.2
# ca-file location for Debian 10
CertificateFile /etc/ssl/certs/ca-certificates.crt

# Remote storage
IMAPStore private-remote
Account private

# Local storage
MaildirStore private-local
Path ~/Mail/private/
Inbox ~/Mail/private/INBOX

Channel private-inbox
Master :private-remote:"INBOX"
Slave :private-local:INBOX
Create Both
Expunge Both

Channel private-drafts
Master :private-remote:"Drafts"
```

```
Slave :private-local:Drafts
Create Both
Expunge Both

Channel private-sent
Master :private-remote:"Sent"
Slave :private-local:Sent
Create Both
Expunge Both

Channel private-trash
Master :private-remote:"Trash"
Slave :private-local:Trash
Create Both
Expunge Both

Channel private-junk
Master :private-remote:"Junk"
Slave :private-local:Junk
Create Both
Expunge Both

Group private
Channel private-inbox
Channel private-drafts
Channel private-sent
Channel private-trash
Channel private-junk
```

Let's look at this in detail:

In the first block we define the new account, give it an alias (»mail« from now on) and provide all credentials we need (host, user and password) and tell msmtp to only connect via a secure connection. The important bits here are the PassCmd and CertificateFile lines:

The PassCmd uses a small shell command to retrieve your password from the systems keychain via the command gpg. While it is called gpg, it is way better than either typing out your password every time or storing it as plain text.

To add your password we executed the command:

```
gpg --encrypt -o ~/.mutt/msmtp-accountname.gpg -r mail@domain.tld
```

which get's read by the line: PassCmd `"echo ${PASSWORD:-$(gpg --no-tty -qd ~/.mutt/msmtp-accountname.gpg)}"`. This way you use your GPG password for all your mail accounts and don't have to memorize them all, or do like me use keepass.

The certificates-file contains all certificates from all the major authorities that mbsync needs for SSL communication. You already have those in your keychain, but mbsync can't just access your keychain, so you have to save those certificates in a single file and point mbsync to it.

Now that we have set up all the credentials and authentification-related stuff, we tell mbsync what to sync where. First up, we define the remote and local storages. You can name them whatever you want, I chose »mail-remote« and »mail-local«. For the local storage we tell mbsync to treat it as a default Maildir. This is the most common format and plays very well with mutt.

The last thing to be done is to provide the so called channels, basically telling mbsync "Hey, if you find that folder on the imap server, please sync it to this local folder on my system". Each folder gets its own channel, Master is the remote-Folder and Slave the local one. Here we use the storage-aliases defined above. The Create/Expunge settings tell mbsync, that it should always delete/create any emails it finds, so it will be totally in sync always. Please note, that you have to be very explicit with the naming of the folders, I had to use the actual value of the broken-umlaut (not UTF-8 here) name of my folder »Entw&APw-rfe« (meaning Entwürfe, which translates to Drafts), to get this to work properly.

Finally you can create a group, that tells mbsync which items should get synced with the command mbsync groupname

I just use one group for each email-account, syncing all its folders. But you could also define something like: sync all inboxes or all archives etc.

Setting up msmtp

For the plain sending of emails, we use msmtp. This basically works the same like setting up mbsync, we start by creating a config file in `~/.msmtprc`, that looks sth. like this:

```
account mail
host mx.nchristian.net
port 587
protocol smtp
auth on
user mail@tinfoil-hat.net
```

```
from mail@tinfoil-hat.net
tls on
tls_starttls on
tls_trust_file /etc/ssl/certs/ca-certificates.crt
passwordeval "gpg2 --quiet --for-your-eyes-only --no-tty --decrypt ~/.config/neomutt/msmtp-mail.gpg"

account private
host mx.nchristian.net
port 587
protocol smtp
auth on
user @nchristian.net
from @nchristian.net
tls on
tls_starttls on
tls_trust_file /etc/ssl/certs/ca-certificates.crt
passwordeval "gpg2 --quiet --for-your-eyes-only --no-tty --decrypt ~/.config/neomutt/msmtp-mail.gpg"
```

Nothing much going on here, you just have to take care that you use the correct SSL-method and port. msmtp can communicate with the. I've added the passwiwordeval command in difference to the original Post since he used MacOS and I am using a minimal Debian, so there's a need for extra asking for a password. The certificates-file is the same file as before.

You can test if this really is working by sending a simple email via the terminal, using this command:

```
echo "Test" | msmtp -a mail youremail@domain.tld
```

Setting up Neomutt

The Neomutt Configuration files are pretty much the same as under MacOS which are explained by the Original Author https://webgefrickel.de/blog/a-modern-mutt-setup-part-two I feel the need to backup his work, in case his Blog goes down, I also contacted the Author on case he want's it to get taken down.

So far we covered installing everything and configuring mbsync and msmtp, for receiving and sending emails. If you did everything correctly, you should now have a folder ~/Mail containing all your emails. So what to do now? -Set up neomutt

Think of mutt as a text-based interface for your email-accounts and the folders therein. You can and should customize almost every keystroke of how you want to use mutt. Have a look at the getting started guide from the neomutt-homepage before diving deeper, especially the ideas of the different »screens« in mutt, for example the index, pager, compose and other screens:

I try to think of those just like the different modes in Vim: you have normal, insert, visual mode etc.—each mode is intended for a specific use case (insert text, edit text etc.) and you can configure different keyboard-shortcuts for every mode. This is basically the same for mutt: each screen has a different purpose (for example the »pager« is for showing email content, the »compose«-screen is for composing messages etc. pp.) and you can configure anything for each mode.

But first we start with some sane defaults for mutt: to bundle everything mutt-config-related I created a folder called ~/.mutt with different config-files. Let's start with the muttrc-file. I gonna split this up, and we will go into the details:

General config

```
# paths
set folder = ~/Mail
set header_cache = ~/.mutt/cache/headers
set message_cachedir = ~/.mutt/cache/bodies
set certificate_file = ~/dotfiles/office/certificates.crt
set mailcap_path = ~/.mutt/mailcap
set tmpdir = ~/.mutt/tmp
```

First, we set the folders and paths for stuff mutt needs. The most important thing here is the first line, pointing mutt to the actual folder where mbsync stores all email. The certificates-file is the same as for mbsync/msmtp as well (see above). We will get to the mailcap-file later… Next up—some sane defaults:

```
# basic options
set wait_key = no
set mbox_type = Maildir
set timeout = 3
set mail_check = 0
set delete
set quit
set thorough_search
set mail_check_stats
```

```
unset confirmappend
unset move
unset mark_old
unset beep_new
```

The most important part here is that we tell mutt to treat our ~/Mail folder as Maildir-format (line 3). This is important, so that mutt can understand the format of the emails we have synced with mbsync. I won't go into the details of every option here, just google for them or have a look at the manual here: neomuttrc manual. Basically this config makes mutt behave: no beeping, sane quitting and real deletion (not archiving) of emails.

Next we configure mutt to behave, when we are in »compose«-view, e.g. when we are writing and sending emails:

```
# compose View Options
set envelope_from                     # which from?
set edit_headers                      # show headers when composing
set fast_reply                        # skip to compose when replying
set askcc                             # ask for CC:
set fcc_attach                        # save attachments with the body
set forward_format = "Fwd: %s"        # format of subject when forwarding
set forward_decode                    # decode when forwarding
set attribution = "On %d, %n wrote:" # format of quoting header
set reply_to                          # reply to Reply to: field
set reverse_name                      # reply as whomever it was to
set include                           # include message in replies
set forward_quote                     # include message in forwards
set editor = "nvim"
set text_flowed
unset sig_dashes                      # no dashes before sig
unset mime_forward                    # forward attachments as part of body
```

I want mutt to ask me if I want to add an CC-field by default (which is something I do very often), show me all headers and just behave nicely when I want to write an email. The important setting here is set `editor = "nvim"` —because that was the whole point of me wanting to use mutt: I want to write emails in Vim.

Moving on: let's configure how everything will be looking and what fields are shown in the overview:

```
# status bar, date format, finding stuff etc.
set status_chars = " *%A"
```

```
set status_format = "[ Folder: %f ] [%r%m messages%?n? (%n new)?%?d? (%d to delete)?%?t? (%t tagged)? ]%>—%?p?( %p postponed )?"
set date_format = "%d.%m.%Y %H:%M"
set index_format = "[%Z] %?X?A&-? %D  %-20.20F  %s"
set sort = threads
set sort_aux = reverse-last-date-received
set uncollapse_jump
set sort_re
set reply_regexp = "^(([Rr][Ee]?(\[[0-9]+\])?: *)?(\^+\] *)?)*"
set quote_regexp = "^( {0,4}[>|:#%]| {0,4}[a-z0-9]+[>|]+)+"
set send_charset = "utf-8:iso-8859-1:us-ascii"
set charset = "utf-8"
```

```
# Pager View Options
set pager_index_lines = 10
set pager_context = 3
set pager_stop
set menu_scroll
set tilde
unset markers
```

```
# email headers and attachments
ignore *
unignore from: to: cc: bcc: date: subject:
unhdr_order *
hdr_order from: to: cc: bcc: date: subject:
alternative_order text/plain text/enriched text/html
auto_view text/html
# when composing emails, use this command to get addresses from
# the addressbook with khard first, and everything else from mu index
set query_command = "( khard email --parsable '%s' | sed -n '1!p'; mu cfind --format=mutt-ab '%s' )"
```

UTF-8 all teh things! Another important line here is the index_format: here we define what fields of an email will be shown in the overview of an email-folder: called the index. We then configure how many lines we want in the pager, basically a condensed overview, when an email is opened and configure everything to prefer text-emails over HTML-emails.

The sidebar

```
# sidebar patch config
set sidebar_visible
set sidebar_short_path
set sidebar_folder_indent
set sidebar_width = 25
set sidebar_divider_char = ' | '
set sidebar_indent_string = '
set sidebar_format = "%B %* [%?N?%N / ?%S]"
```

```
# Mailboxes to show in the sidebar.
mailboxes =ALL-INBOXES
mailboxes =mailbox/INBOX =viu/INBOX
mailboxes ="===================="
mailboxes =mailbox
mailboxes =mailbox/archive =mailbox/sent =mailbox/drafts =mailbox/junk =mailbox/trash
mailboxes =viu
mailboxes =viu/archive =viu/sent =viu/drafts =viu/junk =viu/trash
...
```

The code above configures the sidebar, how wide it should be, what labels are shown etc. pp. I have configured that I first list all of the inboxes from the different mail-accounts (under the label ALL-INBOXES) and then I list all subfolders of the different accounts (two shown here).

The inboxes will all have the same name (INBOX), which is a bit stupid, but I have configured different colors so I know in which inbox I am currently—more on that later.

GPG

GPG/PGP

```
set pgp_sign_as = 2F283D0D
set crypt_use_gpgme = yes
set crypt_autosign = no
set crypt_verify_sig = yes
set crypt_replysign = yes
set crypt_replyencrypt = yes
set crypt_replysignencrypted = yes
```

If you use GPG, set those variables to always sign and encrypt emails if the conversation was started as encrypted. I will not go into detail how to use GPG—that should be up to you. You of course should update the php_sign_as-OPtion to use your GPG-fingerprint.

Colors and key bindings

```
# source colors and keybindings
# keeping those in one place makes it easier for my brain
source ~/.mutt/colors
source ~/.mutt/bindings
```

I've put the key-bindings and color configurations in separate files, to keep it a bit more manageable. You could of course do this with some of the config-parts from above as well.

I won't get into details regarding colors, just use the file from the original authors github repository or search for »mutt color scheme«. The basic idea is: mutt will use your terminal colors and you just tell it what to do and how to color specific things in the different views (pager, index, when composing emails etc.), for example:

```
color index blue default "~N" # new messages
```

will tell mutt to color new messages in the index-view to be blue etc. pp.

And now for the most important part and one of the mightiest features of mutt: key bindings. Good to know: when in mutt you can always press ? to get a list of all available keyboard shortcuts in the current screen, this helps a lot when starting out! By default mutt uses a lot of useful and sane key-bindings, for example, when you are in the index and you have a message selected, hitting d will delete the email, moving it to the trash-folder. I kept a lot's of the default key-bindings from mutt, but also configured some of my own. Let's jump right into my bindings:

```
# some sane vim-like keybindings
bind index,pager k previous-entry
bind index,pager j next-entry
bind index,pager g noop
bind index,pager \Cu half-up
bind index,pager \Cd half-down
bind pager gg top
bind index gg first-entry
bind pager G bottom
bind index G last-entry
```

```
# Sidebar Navigation
bind index,pager <down> sidebar-next
bind index,pager <up> sidebar-prev
bind index,pager <right> sidebar-open
```

Those should be quite easy to grasp: I configure the index and pager views to behave like Vim: j, k, gg, G, Ctrl+u and Ctrl+d for navigation through all emails—just as you are used to from Vim. I use the arrow-keys to navigate in the sidebar. Moving on:

```
# global index and pager shortcuts
bind index,pager @ compose-to-sender
bind index,pager R group-reply
bind index,pager D purge-message
bind index <tab> sync-mailbox
bind index <space> collapse-thread
```

The shortcuts configured in this block will do the following:

- @ with an email selected, it will compose a new email to that emails sender
- D (that is: Shift+d) with an email selected, it will completely delete the email
- R will open Vim with a new »Reply-To-All email«
- hitting Space will collapse threaded emails
- hitting Tab will sync the changes from mutt to your local mailbox. This will not sync remotely to your imap-account, this just propagates changes you did with mutt (deleting, moving emails) to your local mail-folder.

Macros

Those are some basic key-bindings, triggering built-in mutt commands. There a shitload of internal commands you can trigger with just some basic key-bindings, but the real magic makes use of macros—and this is where we marry all those other little tools with mutt:

```
# Save all attachments
macro pager S "<pipe-message> ripmime -i - -d ~/Downloads && rm ~/Downloads/textfile*" "Save
all non-text attachments using ripmime"
# opening urls with urlscan
macro pager \cb "<pipe-message> urlscan<Enter>" "call urlscan to extract URLs out of a
message"
# Sync all email
macro index,pager O "<shell-escape>mbsync -a<enter>" "run mbsync to sync all mail"
```

So, using those bindings hitting S will throw the currently selected message at ripmime (which we've installed in part one) and tell it to save (thus the S) all non-textfile-attachments of the email to my downloads-folder in my home directory.

Ctrl-b will throw the current email (pager only) at urlscan, which will parse the email for any links, so I can easily open them in a browser.

And finally: O (as in »OMG, this works!«) will start mbsync with the configuration from part one and remotely sync all email accounts via IMAP.

So far so good. You should by now have grasped how mutt works and how you can configure it to your liking. I urge you to read the neomutt-introduction (see above) and just hit ? when starting mutt to get an idea of what you can configure. If you only plan on using only one email-account with mutt, you should get pretty far already.

If you plan on using more than one account: get yourself another coffee and read on:

Account-specific configuration with folder-hooks

All the key-bindings and configuration from above are global to mutt, but we still are missing some options to make mutt work correctly with our local email folders and mbsync. If you only want to use one account you should be fine by just setting the options for mbox, trash, from, sendmail etc.—but if you are using more than one account, you have to use something called folder-hooks:

The idea is simple: we change/add configuration for mutt, depending which mail-folder we are currently managing. This way we can use different values for different email-accounts. We need this, because we want to create key-bindings to move emails to corresponding archive-folders, to set different from-email-addresses etc. Have a look at the bottom of the muttrc-file:

by default, use privat

```
set realname = "First Lastname" set spoolfile = "+private/INBOX" source
~/.config/neomutt/accounts/private
```

when changing into other mailboxes, use different adresses etc.

```
folder-hook private/* source ~/.config/neomutt/accounts/private folder-hook mail/* source
~/.config/neomutt/accounts/mail
```

```
set sendmail="/usr/bin/msmtp" # Use msmtp rather than sendmail
```

We set a default and tell mutt to use the account mail and it's configuration (first 4 lines). And then we provide the folder-hooks—this tells mutt: »Hey, if you are doing anything in mail and its subfolders, please use the mail-account-config. If you are doing anything in private and its subfolders, please use the private-account-config« etc. pp.

And with that we are almost done for part 2, let's have a look at the account-specific configs, for example the mail-Config:

```
set from = "mail@tinfoil-hat.net"
set sendmail = "/usr/bin/msmtp -a mail"

# Set folders
set spoolfile = "+mail/INBOX"
set postponed = "+mail/drafts"
set record = "+mail/sent"
set trash = "+mail/trash"

# custom signaure
set signature = ~/.mutt/signatures/tinfoil

color status cyan default

macro index o "<shell-escape>mbsync mailbox<enter>" "run mbsync to sync mail for this account"

macro index,pager J \
  "<enter-command>set my_old_resolve=\$resolve noresolve<enter>\
  <tag-prefix><clear-flag>n<enter-command>set resolve=\$my_old_resolve<enter>\
  <save-message>+mail/junk<enter>" \
  "mark as read and move to junk folder"

macro index,pager I \
  "<save-message>+mail/INBOX<enter>"  \
  "move message to the inbox"
```

OK, let's break this down line by line: set from tells mutt which email to use as the from-email—this should match the one from the mbsync-config from part one. set sendmail tells mutt what program it should use to send emails, and in this case: we use msmtp and tell msmtp to use the config for the mailbox-account. This is where all the tools from part one are married with mutt. We then tell mutt which folders to use for what (inbox to trash), set additional email-addresses we want to use for this account and use a custom signature with set signature.

color status cyan default sets the color of the status-bar, and by setting a different color in each account-config you can visually differentiate your email-accounts. Nice.

After having set everything mutt needs to read folders and send emails correctly, I then configure some more macros, that are specific to the currently active account:

- o will start mbsync to sync all folders via IMAP (for the mailbox-account only)
- J will move a selected email to the spam-folder in the mailbox-account
- A will archive an email in the mailbox-account (move it to mailbox/archive)
- I will move an email back to the inbox (e.g. when browsing the archive or sth.)

Those macros and settings will be different for every account-config-file have a look here, but in the end do the same actions—but account-specific.

So with folder-hooks I can just memorize one keyboard-shortcut (J for Junk), but it will work in every account and I don't have to care about setting the correct from-email manually. When sending an email from my viu-inbox, it will set the correct option with the folder-hook automatically.

Phew. Still here? Still following? Awesome—we are halfway there. Step 4: Using mutt

So let's recap: in part one we configured msmtp and mbsync to work with our email-accounts. In part two we learned what mutt is for, and how we can configure it to our liking. With key-bindings, macros and folder hooks we just managed to marry mutt with everything from part one, so we now should be able to send and receive emails from within mutt!

Instead of trying to describe how my workflow looks like I will just show it to you, brace for some video-content! (This is the first time I am ever doing this, so please bare with my awkardness. Thanks.)

https://webgefrickel.de/content/3-blog/21-a-modern-mutt-setup-part-two/using-mutt.mp4

Other Sources

https://neomutt.org/man/neomuttrc

https://neomutt.org/guide/gettingstarted

https://github.com/webgefrickel/dotfiles/blob/master/office/

https://wiki.archlinux.org/index.php/Msmtp

Changes include the adoption to a Debian based Operating System and the Configuration of my own E-Mail Server