Reading Email in Emacs with mu4e

David Florness

February 14th, 2019
Overview

- A brief history of email in Emacs
- What’s mu4e?
- Why mu4e?
- offlineimap
- smtp
- mu{,4e} Installation
- config
- Emacs packages config
- mu4e config
- Signing and encrypting with PGP
- Demo
A brief history

- Reading and writing email in Emacs has been a thing for a while.
- Gnus, a “news” and (now) email client, has shipped with Emacs since November 1995.
- “News” in this sense means content from the Usenet network, which is a discussion system conceived in 1979 that uses the UUCP (Unix-to-Unix Copy) suite of protocols.
- I’m not a fan because
  - the documentation is old and clunky
  - messages can only be read once, and changing this is an “uphill battle”:

  Gnus does not behave like traditional mail readers. If you want to make it behave that way, you can, but it’s an uphill battle.¹

¹from the Gnus manual, section 6.4.1
Introducing **mu** and **mu4e**

- Created by Dirk-Jan Binnema, djcb:
  - **mu** is a tool for dealing with e-mail messages stored in the Maildir-format, on Unix-like systems. **mu**’s main purpose is to help you to find the messages you need, quickly.²
  - **mu4e** is simply a front end for Emacs (**mu4e** := mu 4 (for) Emacs)

² from the **mu** website, [http://www.djcbsoftware.nl/code/mu/](http://www.djcbsoftware.nl/code/mu/)
Why \textit{mu}? 

- From the \texttt{mu4e} manual\textsuperscript{3}, section 1.1:
  \begin{quote}
  I (the author) spend a \_lot\_ of time dealing with e-mail, both professionally and privately. Having an efficient e-mail client is essential. Since none of the existing ones worked the way I wanted, I thought about creating my own.
  \end{quote}

- Once you get used to it, \texttt{mu4e} makes reading and writing email less painful (might I say fun?)
To get started, the first step is to download all of your emails using the Internet Message Access Protocol (IMAP); I highly discourage POP3!

If you have a Gmail account (which you do, since it's what the Mines MyMail accounts are) and want to use it, you'll need to enable "Insecure Apps".

For Gmail, also create an application-specific password. (recommended)
~/.offlineimaprc

[general]
accounts = mines
pythonfile = ~/.offlineimap.py

[Account mines]
localrepository = mines-local
remoterepository = mines-remote

[Repository mines-local]
type = Maildir
locafolders = ~/mail/mines

[Repository mines-remote]
type = Gmail
remoteuser = davidflorness@mymail.mines.edu
remotepasseval = get_mines_pass()
sslcacertfile = /etc/ssl/certs/ca-certificates.crt
ssl_version = tls1_2
ssl = yes
#!/usr/bin/env python2
from subprocess import check_output
from getpass import getpass

def get_mines_pass():
    try:
        return check_output("pass mail/mines/imap", shell=True).splitlines()[0]
    except:
        return getpass("Mines-imap password: ")
Time to download your email!

- With the aforementioned configs in place, just run the following to download your email:
  
  offlineimap -o

- This will take a while the first time...
Next, you need to set up some way to send your email. 99% of the time, this is done with the Simple Mail Transfer Protocol (SMTP). 

`msmt`p is a very simple STMP client that sends your email and nothing more.
# Always use security
defaults
 auth on
tls on
tls_trust_file /etc/ssl/certs/ca-certificates.crt
 protocol smtp

# College account
account college
host smtp.mines.edu
port 587
from davidflorness@mines.edu
user davidflorness
password eval "pass show multipass"
I know no one needs to this, but just in case:

```
pacman -S emacs
```

And install mu itself (this comes with mu4e, as well)

```
pacman -S mu
```

Of course, swap `pacman` with whatever your distro uses (`apt`, `yum`, ...) and look up what the package names actually are. If no package for `mu` exists for your distro, you can install from source (or switch to a better distro).
Tell `mu` where your Maildir is:

```bash
mu index --rebuild --maildir ~/mail
```
(require 'package)

;;; Add org and melpa package archives
(let* ((no-ssl (and (memq system-type '(windows-nt ms-dos))
                     (not (gnutls-available-p))))
       (melpa-url (concat (if no-ssl "http" "https")
                          "://melpa.org/packages/"))
       (org-url (concat (if no-ssl "http" "https")
                        "://orgmode.org/elpa/"))
       (add-to-list 'package-archives (cons "melpa" melpa-url))
       (add-to-list 'package-archives (cons "org" org-url)))

;;; Load and activate lisp packages
(package-initialize)

;;; Fetch the list of available packages
(unless package-archive-contents
  (package-refresh-contents))

;;; Install use-package for easy package configuration
(unless (package-installed-p 'use-package)
  (package-install 'use-package))

(require 'use-package)
(use-package mu4e
  :defer t
  :commands (mu4e mu4e-compose-new)
  :config
  (progn
    (frypan/setup-mu4e)
    (add-hook 'mu4e-compose-mode-hook 'frypan/mu4e-encryption)))

(use-package evil-mu4e
  :ensure t
  :after mu4e)
(defun frypan/setup-mu4e ()
  (setq user-full-name "David Florness"
        mu4e-compose-signature "David

Sent from my Emacs"

;;; SMTP
    message-send-mail-function 'message-send-mail-with-sendmail
;;; send mail using address in message
    message-sendmail-extra-arguments '("--read-envelope-from")
    message-sendmail-f-is-evil 't
    sendmail-program "msmtp"

;;; mu4e
    mu4e-maildir "~/mail"
    mail-user-agent 'mu4e-user-agent
    message-kill-buffer-on-exit t
    mu4e-view-show-images t
    mu4e-view-show-addresses t
    mu4e-get-mail-command "offlineimap -o"
    mu4e-headers-include-related nil)
Setup functions II

```lisp
;; contexts (i.e. accounts)
mu4e-contexts
  ; (make-mu4e-context
    :name "college"
    :match-func (lambda (msg)
      (when msg
        (or
          (mu4e-message-contact-field-matches msg
            :to "davidflorness@mines.edu")
          (mu4e-message-contact-field-matches msg
            :to "davidflorness@mymail.mines.edu"))
        )))
  :vars '((mu4e-trash-folder . "/mines/[Gmail].Trash")
         (mu4e-drafts-folder . "/mines/[Gmail].Drafts")
         (mu4e-sent-folder . "/mines/[Gmail].Sent Mail")
         (mu4e-sent-messages-behavior . sent)
         (user-mail-address . "davidflorness@mines.edu")
         (mu4e-maildir-shortcuts
          ("/mines/INBOX" . ?i)
          ("/mines/[Gmail].Sent Mail" . ?s)
          ("/mines/[Gmail].Trash" . ?t)
          ("/mines/[Gmail].All Mail" . ?a)
          ("/mines/[Gmail].Drafts" . ?d)
          ))))
```

David Florness
;;; setup encryption for mu4e
(defun frypan/mu4e-encryption ()
  ;; always sign messages
  (mml-secure-message-sign-pgpmime)

  (let ((msg mu4e-compose-parent-message))
    (when msg
      (when (member 'encrypted (mu4e-message-field msg :flags))
        ;; encrypt message if replying to encrypted message
        (mml-secure-message-encrypt-pgpmime))))
Starting mu4e

- And your configuration is all done!
- Now just start mu4e with
  
  M-x mu4e
Demo time!