Snooze via IMAP
hide this email until later
Message Snooze

rationale

• Snooze allows a message to be filed away for a preset period of time so that its presence in a message list will not distract the user.

• Messages can still be found and unsnoozed or re-snoozed.

• Messages are "awakened" by the server and put back in place.

• A newly-awakened message can be distinguished from a newly-delivered one.
Message Snooze
protocol interface

• UID SNOOZE uid until-time [ (add-flags) [ (remove -flags) ] ]
  • if currently selected mailbox is not \Snoozed mailbox
    • message is MOVE-d to the mailbox with the special use \Snoozed
    • the message's AWAKEN-TO data item is set to the currently selected mailboxid
    • the message's AWaken-FLAGS data item is set
    • the message's AWaken-AT data item is set
  • if currently selected mailbox is \Snoozed mailbox
    • the message's AWaken-FLAGS data item is updated
    • the message's AWaken-AT data item is updated
Message Snooze
protocol interface

• at AWAKE-AT, the server:
  • moves the message to its AWAKEN-TO origin
  • if the AWAKEN-TO mailbox no longer exists, move to INBOX
  • updates the message flags as per AWAKE-FLAGS
Message Snooze
protocol interface

- UID SNOOZE uid until-time
  - message is MOVE-d back to AWAKEN-TO mailbox
  - AWAKEN-AT is cleared
  - AWAKEN-FLAGS are cleared
  - (this one might be controversial! 😊)
Message Snooze
the \Snoozed mailbox

- the user may not move or append messages to this mailbox
- the user may not MOVE mail out of this mailbox
- the mailbox is created as needed (when SNOOZE occurs and \Snooze does not exist)
Message Snooze for greater success

- Fastmail's existing implementation uses $new flag to indicate an awoken message that has not yet been re-seen.
- The SAVEDATE extension is convenient to determine when a message with $new was awoken.
Message Snooze

see also

- Sieve Email Filtering: Snooze Extension
  - draft-ietf-extra-sieve-snooze
IMAP Mailboxes as Labels

show me all this email's mailboxes
Labels: Get a Message's Labels
fetching mailbox list

• the server must implement RFC 8474, OBJECTID capability
• a new data item, MAILBOXIDS is added to messages
• MAILBOXIDS is a list of the mailbox ids of mailboxes in which this message appears
• MAILBOXIDS is read only, and is updated with COPY, MOVE, DELETE, &c.
Labels: Get a Message's Labels
fetching mailbox list

• complication: modseq when MAILBOXIDs change; example:
  • email E1 is in mailbox M1 as uid U1; in mailbox M2 as U2
  • both U1 and U2 have been cached along with the MAILBOXIDS data item
  • while M2 is selected, U2 is deleted and expunged
  • this should change the MODSEQ of U1
Labels: Set a Message's Labels

- nothing to it: DELETE (and expunge) from mailbox to remove label
- COPY to mailbox to add label
  - (we want COPY-IF-NOT-PRESENT)
Labels: Mailbox Color

- we define a metadata name for color, /shared/color
- color must be a css3 color name or #rrggbb color code
- a LIST extension is adopted for fetching metadata in LIST
- when bootstrapping a new client:
  - LIST, fetching color and mailboxid
  - SELECT INBOX, fetch mail with MAILBOXES
- labels can be drawn